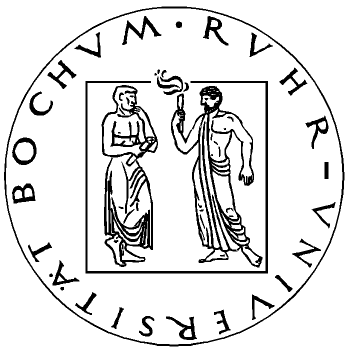
DES Sidechannel Collision Attacks On Smartcard Implementations

Diplomarbeit

by

Kai Schramm

Department of Electrical Engineering and Information Sciences
Ruhr-Universität Bochum
Communication Security (COSY) Group



Advisors: Prof. Dr. Christof Paar, Thomas Wollinger, M.S.

Beginning: February 8th 2002
End: August 7th 2002

Erklärung

Hiermit versichere ich, dass ich meine Diplomarbeit selbst verfaßt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Ort, Datum _____

Acknowledgements

I would like to thank Prof. Dobbertin for his initial idea of a collision attack against DES. Also, I want to thank the entire group for Communication Security at the Ruhr Universität Bochum. A special thank-you goes to Thomas Wollinger and Prof. Christof Paar for their ideas and inspirations.

Above all, I want to thank my parents for their love and support.

Abstract

Until now in cryptography the term collision was mainly associated with the surjective mapping of different inputs to an equal output of a hash function. Previous collision attacks were only able to detect collisions at the output of a particular function. In this thesis a new class of attacks is introduced which uses side channel analysis to detect internal collisions. We applied our attack against the widely used Data Encryption Standard (DES). We show that internal collisions can be caused in the S-Boxes of DES in order to gain information about the secret key-bits. As result, we were able to exploit an internal collision with a minimum of 140 encryptions yielding 10.2 key-bits[1]. Moreover, we successfully applied the attack to a smart card compatible processor.

[1] averaged over 10,000 keys

Contents

1 Introduction **7**

2 Previous Work **9**
2.1 Side Channel Attacks Against DES 9
2.2 Cryptanalytic Attacks against DES 13
2.3 Collision Attack against COMP128 15

3 DES Collision Attack **19**
3.1 Introduction to DES 19
3.2 Wiemers' collision attack against DES 22
3.3 Collisions in a Single S-Box 34
3.4 Collisions in Two S-Boxes 38
3.5 Collisions in Three S-Boxes 39

4 Optimization of the Collision Attack **52**
4.1 Multiple Differentials 54
4.2 Linear and Stochastical Dependencies 59
4.3 Key Candidate Reduction 71

5 An Implementation of the Attack **73**
5.1 Computer Simulation 73
5.2 Measurement Equipment 73
5.3 Compromising DES on an 8051 Microcontroller 78
5.4 Hardware Countermeasures of a Secure Smartcard 81

6 Results and Conclusions **83**
6.1 Results of our DES collision attack 83
6.2 Comparison with Wiemers' attack 85
6.3 Future Work 86

A Appendix **88**
A.1 DES S-Boxes 88
A.2 S-Box δ-tables 90
A.3 S-Box Δ-tables 105

1 Introduction

This thesis introduces a new attack on the Data Encryption Standard (DES) block cipher. Randomly chosen plaintexts[1] are input to DES until an internal collision occurs. In cryptography a collision is defined as the surjective mapping of two different inputs to one equal output of a particular function. In general, hash functions are associated with collisions, however, any cipher which contains surjective substitution boxes such as DES or COMP128 can cause collisions within the algorithm.

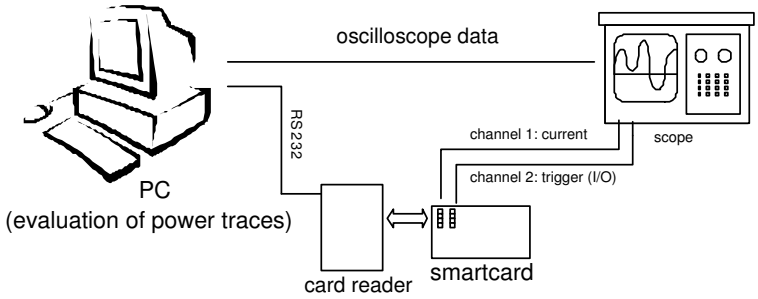


Figure 1.1: Typical measurement setup for power analysis

Due to the design of DES, internal S-Box collisions can not be detected at the output of the algorithm. However, a possibility to detect internal collisions is power analysis of the target implementation. Thus, the attack combines DES cryptanalysis with side channel evaluation. A typical example of a target implementation is a smartcard microprocessor. A microprocessor generally yields side channel information such as timing behaviour, electromagnetic emanation and electrical power consumption. Measuring the latter one, Paul Kocher showed in 1998 how electronic devices can be broken [KJJ99, KJJ98] using the *simple power analysis* (SPA) and the *differential power analysis* (DPA). A typical measurement

[1] a ciphertext attack is possible as well

setup for power analysis is shown in Figure 1.1. The measurement setup consists of an oscilloscope sampling the power consumed by the cryptographic device and a computer analyzing the measured power traces and triggering new measurements.

In this thesis it is shown that a collision within three adjacent S-Boxes of round one of DES can be found with as few as 140 encryptions[2] exposing detailed information of 18 bits of the secret DES key. A collision is detected by correlating the power traces of different DES encryptions using the measurement setup shown in Figure 1.1.

The remaining chapters of this thesis are organized as follows:

In Chapter 2 previous attacks on DES such as SPA, DPA, linear and differential cryptanalysis are reviewed. Also, the well-known collision attack against the GSM authentication algorithm COMP128 is illustrated.

Chapter 3 first reviews the fundamentals of DES. Then, the alternative DES collision attack by Andreas Wiemers and Prof. Hans Dobbertin of the *Bundesamt für Sicherheit in der Informationstechnik* (BSI) is explained. In general, the Wiemers attack is a very pragmatic attack against cryptographic algorithms based on the Feistel cipher, while our collision attack focuses on the the S-Boxes of DES.

The optimization of the DES collision attack is presented in Chapter 4. It is shown how linear and stochastic dependencies within the S-Boxes can be used to minimize the measurement costs of power analysis. This chapter also describes how the number of key candidates can be delimited using further collisions.

In Chapter 5 the DES collision attack against an 8051 compatible microcontroller running DES in assembler is successfully demonstrated. Moreover, the difficulties of the attack against a state of the art smartcard processor with hardware countermeasures are explained and possible solutions for a successfull attack are given.

In Chapter 6 the results of the optimized attack are shown and comparison to the Wiemers attack is given. Moreover, additional research topics related to the attack are given.

Finally, the Appendix lists the S-Boxes of DES and the derived differential tables used for the collision attack.

[2]averaged over 10,000 attacks with random keys

2 Previous Work

2.1 Side Channel Attacks Against DES

Cryptographers have traditionally designed new cipher systems in assumption that the system would be realized in a closed, reliable computing environment which does not leak any information about the internal state of the algorithm. In electrical engineering such an ideal mathematical object would equal a black box with an input and an output interface. However, in the last years it has become clear that any implementation of a cryptographic system can leak information about the ongoing operations it processes. The term *side channel* describes this originally unwanted source of information.

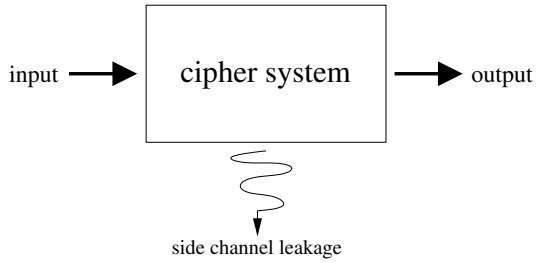


Figure 2.1: Information leakage through a side channel of a cryptographic system

In general, a cipher system is physically implemented as a microchip made of semiconductor logic gates and storage elements. Such a system will yield different side channel information such as timing measurements [Koc96], power consumption [KJJ98, KJJ99, AO, MDS99, MS00] and electromagnetic emission [AK96]. It is known that sophisticated smartcard companies are tyring to eliminate side channel information as much as possible, e.g., by means of hardware and software countermeasures [CCD00]. However, due to the nature of its circuit implementation, cipher systems will always provide side channel information, which eventually can be used to exploit the system.

In [KJJ98] Paul Kocher *et al.* showed that common ciphers such as DES and RSA can be broken by analyzing the power consumption of the target implementation. A cipher system embedded into a microchip usually consists of many thousand *Complementary Metal Oxid Semiconductor* (CMOS) logic gates, which are constructed out of transistors. In Figure 2.2 a simple CMOS inverter consisting of a *p-channel Metal Oxid Semiconductor* PMOS and an *n-channel Metal Oxid Semiconductor* NMOS transistor is shown.

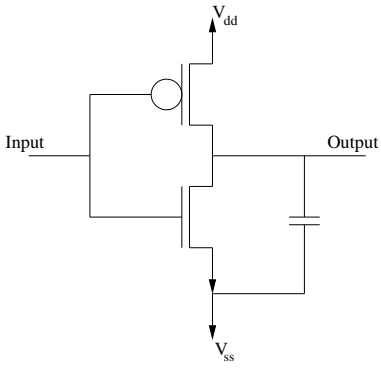


Figure 2.2: CMOS inverter

The power consumption of a microchip is analyzed by putting a small shunt resistance R_s (e.g., 10Ω) between the V_{dd} (V_{ss}) pin of the microchip and the true source (ground).To reduce noise induced by the power source, it should be replaced by a well-filtered source with low voltage ripple. A digital oscilloscope with low quantization noise (e.g., 12 bit *Analog Digital Converters* ADCs) and high sampling rate (e.g., 1GHz) is then used to digitize the voltage over the shunt resistance, which is proportional to the current drawn by the microchip. This is shown in Figure 2.2.

Since a second channel of the oscilloscope might be used for triggering, it can be advantegous to measure the current between the V_{ss} pin and ground, because all channels of the scope can then use ground as a common reference voltage.

Power analysis can be classified into two general methods: *Simple Power Analysis* (SPA) and *Differential Power Analysis* (DPA). In SPA related attacks an adversary measures the power consumption and thus deduces information about specific instructions and their operands on the data bus. SPA measurements are strongly hardware dependent. Moreover, the exact point of time of the observed instruction must be known. In [MDS99] it is stated that generally two types of information leakage have been observed in SPA: Hamming weight leakage and

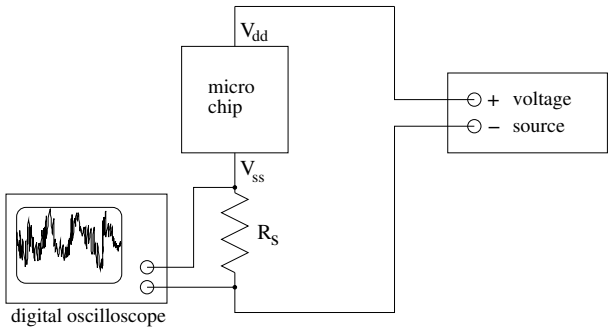


Figure 2.3: Power analysis of a micro chip

transition count leakage. In a precharged bus design [WE93] the number of zeros driven onto the bus is directly proportional to the amount of current that is being discharged from the driven gates. Thus it is possible to determine the Hamming weight on the bus. Transition count information leaks when the dominant source of current is due to the switching of the gates that are driven by the bus: during a gate transition from high to low or low to high both transistors in Figure 2.2 will conduct current for a short time. This transition current consists of two parts [AO]: a larger part, which arises from charging/discharging succeeding gates and parasitic capacitances, and a smaller part, which is due to the dynamic short circuit current between V_{dd} and V_{ss}. Theoretically, it is possible to distiguish between an output state change from high to low and low to high, because discharging succeeding gates will result in an increased transition current, while charging will result in a decreased transition current [MDS99].

As explained above, SPA directly examines single power traces. Nevertheless, an attacker must have detailed information about the hardware and about the particular algorithm implementation. If attacking DES for example, an attacker could analyze the power traces during the PC1 permutation in the key scheduling algorithm. It is assumed that the attacker is able to determine the Hamming weight of each key byte by measuring the peak height of key byte related fetch instructions. This would then provide [Mui01]

$$-\sum_{i=0}^{8} \frac{\binom{8}{i}}{2^8} lg \frac{\binom{8}{i}}{2^8} \approx 2.54 bits \tag{2.1}$$

of information for each key byte. The size of the key space would thus be reduced to $56 - 7 \times 2.54 \approx 38$ bits making DES very vulnerable to brute force attacks. In [MDS99] an even more powerful Hamming weight attack against the 16 DES

subkeys[1] is explained. Once the Hamming weight of each of the 6 bytes of all round keys is determined, an attacker simply has to solve a matrix of $16 \times 6 = 96$ equations with 56 unknowns in order to determine the secret DES key.

Other occasions where implementations of cipher algorithms might be vulnerable to SPA are carry bit related instructions, such as shifting the key bytes or the use of conditional branches to test bit values [MDS99].

DPA, also invented by Kocher *et al.* [KJJ99], is more complex, but also more powerful than SPA, because it has a few important advantages: no specific information about the analyzed hardware and the cryptographic instructions needs to be known, uncorrelated (white) noise superposed to measurements will be filtered out and no knowledge about time offsets is required.

Once again, an implementation of DES shall be compromised. An adversary must be able to encrypt (decrypt) arbitrary plaintexts (ciphertexts) in order to determine bits of the first (last) DES subkey. Here, an attack on the last round key is assumed. A selection function $D(C, b, K_s)$ defines the value of the bit $0 \leq b \leq 31$ of the 32 bit output of the 8 S-boxes in round 16 when decrypting ciphertext C. The input of the corresponding S-box belonging to bit b is x-ored with $0 \leq K_s \leq 2^6$ of the last subkey. The adversary chooses b and K_s, encrypts n plaintexts and computes $D(C, b, K_s)$ for each encryption. If $D(C, b, K_s)$, i.e., the value of bit b, equals one, the power trace T_i of the encryption is added to a differential trace Δ_D. If $D(C, b, K_s)$ equals zero, the power trace T_i is subtracted from Δ_D:

$$\Delta_D = \frac{\sum_{i=1} nD(C, b, K_s)T_i}{\sum_{i=1} nD(C, b, K_s)} - \frac{\sum_{i=1} n(1 - D(C, b, K_s))T_i}{\sum_{i=1} n(1 - D(C, b, K_s))} \tag{2.2}$$

If K_s is chosen incorrectly, the selection function $D(C, b, K_s)$ is uncorrelated to the actual computation of the DES implementation and Δ_D will diminish to zero:

$$\lim_{n \to \infty} \Delta_D \approx 0 \tag{2.3}$$

However, if K_s is chosen correctly, the selection function $D(C, b, K_s)$ is completely correlated with those instructions operating on the target bit b. This will be indicated by distinct peaks in the differential trace. Moreover, the time offsets of the peaks reveal at what points of time instructions operate on bit b.

[1] synonymous with round key

2.2 Cryptanalytic Attacks against DES

Since DES had become an encryption standard a number of attack methos have been developed: differential cryptanalysis, linear cryptanalysis and brute force. The history of DES cryptanalysis is shown in Table 2.1.

Date	Proposed/implemented attack
1977	Diffie & Hellman, estimate cost of key search machine (underestimate)
1990	Biham & Shamir propose differential cryptoanalysis (2^{47} chosen ciphertexts)
1993	Mike Wiener proposes detailed hardware design for key search machine: average search time of 36 h @ $100,000
1993	Matsui proposes linear cryptoanalysis (2^{43} chosen ciphertexts)
Jun. 1997	DES Challenge I broken, distributed effort took 4.5 months
Feb. 1998	DES Challenge II–1 broken, distributed effort took 39 days
Jul. 1998	DES Challenge II–2 broken, key-search machine built by the Electronic Frontier Foundation (EFF), 1800 ASICs, each with 24 search units, $250K, 15 days average (actual time 56 hours)
Jan. 1999	DES Challenge III broken, distributed effort combined with EFF's key-search machine, it took 22 hours and 15 minutes.

Table 2.1: History of full-round DES attacks [Paa02]

In the year 1999 Biham and Shamir introduced the differential cryptanalysis [BS91] as a new generic attack against iterated block ciphers. Differential analysis involves comparing the x-or (exclusive-or) of two plaintexts to the x-or of the corresponding two ciphertexts. Two plaintexts can be chosen at random, as long as they satisfy a particular x-or difference. Certain differences, called "characteristics", in the plaintext pairs have a high probability of causing certain differences in the resulting ciphertext pairs. Characteristics extend and define a path through several rounds. The distribution of characteristics can be found by generating a table where the rows represent the possible input differences, the columns represent the possible output differences, and the entries represent the probability of this particular characteristic. A plaintext pair that satisfies the characteristic is a correct pair, the pair that does not is a wrong pair. A correct pair will suggest the correct round key (for the last round of the characteristic), a wrong pair will suggest a random key. Collecting enough guesses will yield one subkey that will rise out of all the random alternatives. Biham and Shamir were able to prove that the secret key can be determined by analysis of 2^{47} chosen

plaintext - ciphertext DES pairs or analysis of 2^{55} known plaintext - ciphertext DES pairs. Although DES can theoretically be broken by differential analysis, this attack is not practical, because of the large data requirements. Moreover, this is an indication that the S-Boxes in DES were optimized against differential analysis - 15 years before the attack became publicly known.

Linear cryptanalysis was invented in 1993 by Matsui [Mat94]. Its goal is to statistically approximate a linear relationship between certain plaintext, key and ciphertext bits

$$P[i_1, i_2, ..., i_a] \oplus C[j_1, j_2, ..., j_b] = K[k_1, k_2, ..., k_c]^2 \tag{2.4}$$

where $P[i]$ denotes the x-or sum[3] of chosen plaintext bits, $C[j]$ denotes the x-or sum of chosen ciphertext bits and $K[k]$ denotes the x-or sum of chosen key bits. A probability p can be associated with each linear relationship. A probability p of 0.5 states that no linear relationship exists for the plaintext bits and ciphertext bits specified. However, the more p deviates from 0.5, the better this linear relationship predicts the parity of key bits involved. The more plain- and ciphertexts are analyzed, the more reliable is the approximation. Since all operations in DES, except the S-boxes, are linear, it suffices to derive linear approximations of input and output bits of the S-boxes using the Walsh transform [Dob01]. These derived relationships can then be extended to all 16 rounds, thus describing the full cipher (except for the initial and final permutation, which does not improve DES security). In 1994 Matsui improved linear cryptanalysis against DES. The final result was that DES is breakable by linear cryptanalysis at a success rate of 85% if 2^{43} plaintexts ciphertext pairs (ca. 64,000 GB of data) are known. Matsui *et al.* successfully carried out an attack against DES in 1993 using twelve HP9000 99MHz workstations.

Both differential and linear cryptanalysis require huge amounts of data. However, an exhaustive search for the key using highly parallelized hardware, such as clusters or custom hardware, does not need mass storage and might be even faster. DES key length is 56 bits, i.e., there exist 2^{56} possible keys. Exhaustive search for the secret key with one known plaintext-ciphertext pair thus requires 2^{56} DES operations.

RSA Data Security [RSA] initiated the first DES challenge in 1997 offering $10,000 to the first person giving them the correct key of a known plaintext

[2] the $\oplus$ operator denotes the binary x-or operation throughout this thesis
[3] synonymous with *parity*

- ciphertext pair. Rocke Verser, a contract programmer, created an exhaustive search system on a client/server basis in his spare time. Many volunteers in the internet installed his client software. At its peak 7 billion keys were tested per second [RSA]. The correct DES key was finally found in June 1997. The total effort took about four and a half months.

In February 1998, distributed.net solved RSA's DES challenge II-1 using an internet cluster[4] of about 50,000 voluntary processors donating idle time. The cluster searched 85% of the possible keys in 41 days.

In July 1998, RSA's DES challenge II-2 took place. The *Electronic Frontier Foundation* (EFF) designed the supercomputer *DES Cracker*. It was built out of 1800 custom microchips called *DEEP Crack* and total costs were about $250,000. DES Cracker was able to find the secret DES key within 56 hours testing up to 92 billions of keys per second.

Finally, the last DES challenge III happened in January 1999. In a combined effort DES Cracker and a cluster of 100,000 computers[5] broke the DES within 22 hours testing up to 245 billion keys per second [RSA].

2.3 Collision Attack against COMP128

The alogrithm COMP128 was initially suggested by the *Group Speciale Mobile* (GSM) commitee. It represents a possible implementation of the authentication algorithms A3 and A8, which are based on the challenge/response principle. Technical details of COMP128 were strictly confidential, however the algorithm was completely reverse engineered by M.Briceno *et al.* [BGW98a] in 1998. The basic structure of COMP128 is shown in Figure 2.4.

The input of COMP128 is a concatenation of a secret key K_i (16 bytes), which is stored in the SIM card, and a random number $RAND$ (16 bytes) resulting in a 32 byte number x [Zen99]. This number is run through a hash function, which maps x (32 bytes) to x' (16 bytes) using substitution boxes. The resulting number x' is then permutated and again concatenated with K_i to 32 bytes. These resulting 32 bytes are again input to the hash function. After nine iterations (see Figure 2.4) the output of the hash function x' is the final output of COMP128. The 16 output bytes contain a session response number SRES (4 bytes) used

[4]assisted by *distributed.net*
[5]assisted by *distributed.net*

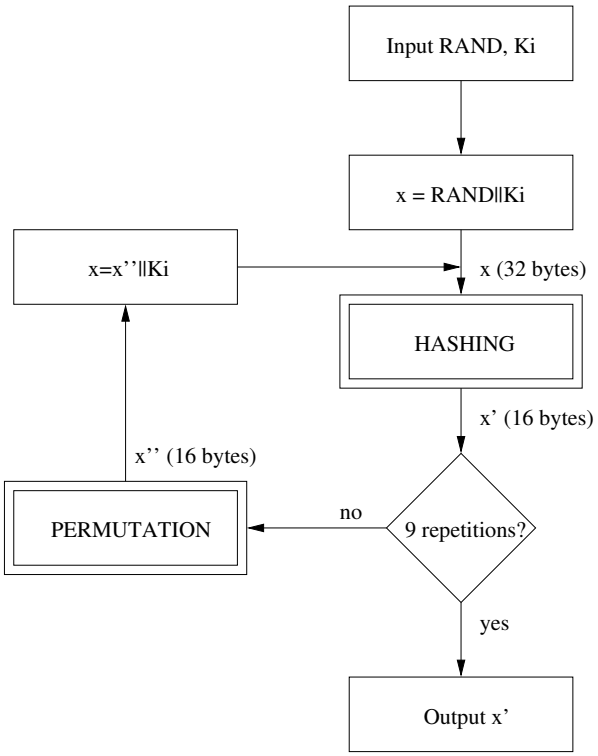


Figure 2.4: Schematics of COMP128

for authentication of the mobile phone (i.e., its SIM module) with the network provider and a session key K_c (8 bytes) used for the stream cipher A5.

The structure of the hash function is shown in Figure 2.5. It is based on the butterfly structure and consists of five rounds. The S-boxes in rounds 1,2,3,4 and 5 substitute the input by 8,7,6,5 and 4 bit values, respectively.

The lower 16 bytes that are input to the COMP128 represent the secret key K_i. These bytes are fixed and cannot be altered by an adversary. The upper 16 bytes contain the random number $RAND$. These bytes can be varied. Each S-box in the first round of the hash function maps one input byte from $RAND$ and one input byte from K_i to one output byte, i.e., $f : 2^8 \times 2^8 \rightarrow 2^8$. It can be shown by computer simulation that the S-boxes in the first round simply permutate[6] the input byte belonging to $RAND$. Therefore, it is not possible to cause collisions in the first round [Zen99]. However, collisions can occur in round 2. A collision attack against COMP128 was first demonstrated by M.Briceno *et al.* in 1998 [BGW98b]. Pairs of S-boxes (see Figure 2.5) in round 2 map $2^8 \times 2^8$

[6] which is a bijective mapping

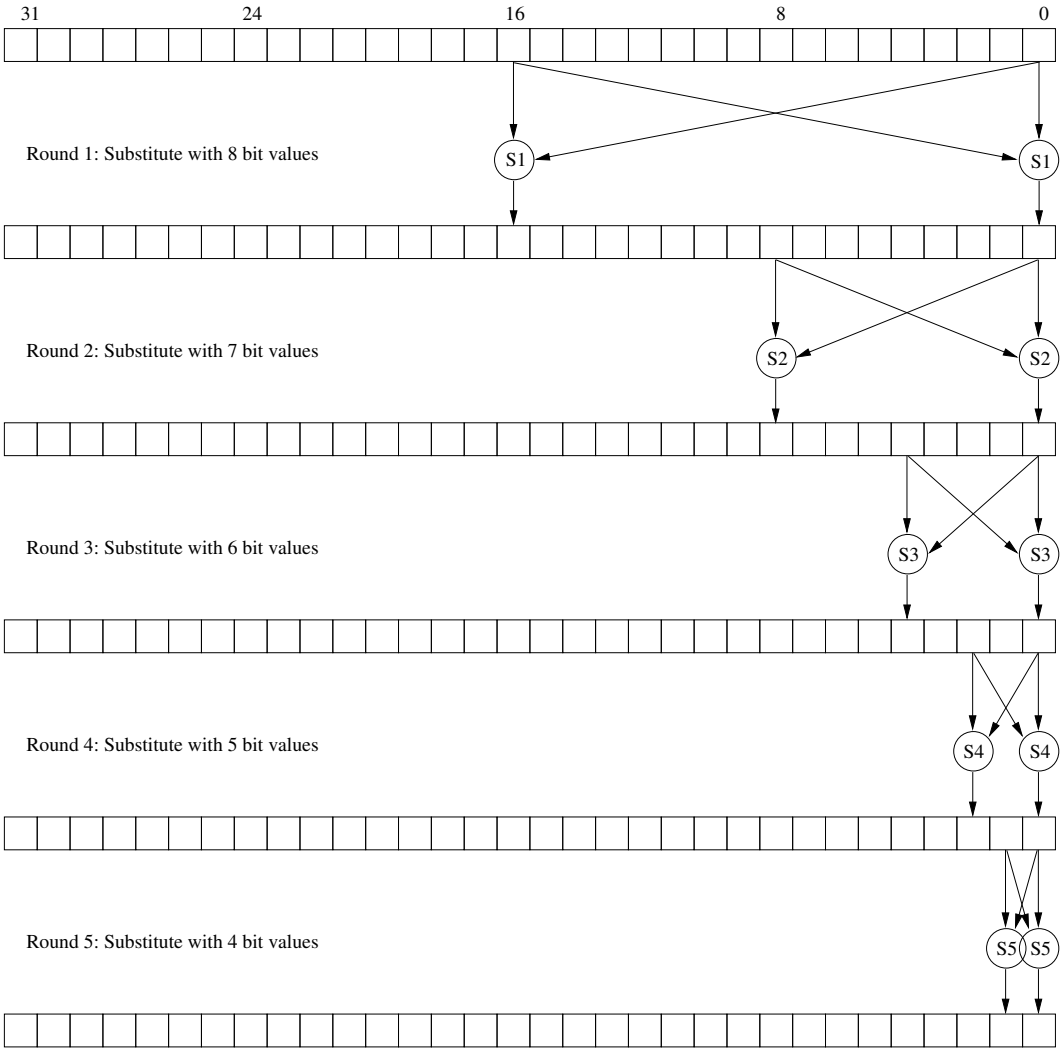


Figure 2.5: Schematics of COMP128's hash function

values surjectively to $2^7 \times 2^7$ output values. This mapping is uniformly distributed, i.e., four different inputs are mapped to one output. Thus, every 7 bit output of round 2 is a function of four input bytes of round 1. This is shown in Figure 2.6. In round 5, each output nibble (4 bits) is a function of all 32 input bytes of round 1.

If a collision occurs in round 2, it will propagate through all following rounds and iterations of the hash function. As a result, a collision in COMP128 will be detectable at the output[7]. An adversary simply needs to vary two bytes of

[7]It must be pointed out here that a collision inside a round of DES can not be detected at the output, because DES is based on the Feistel cipher

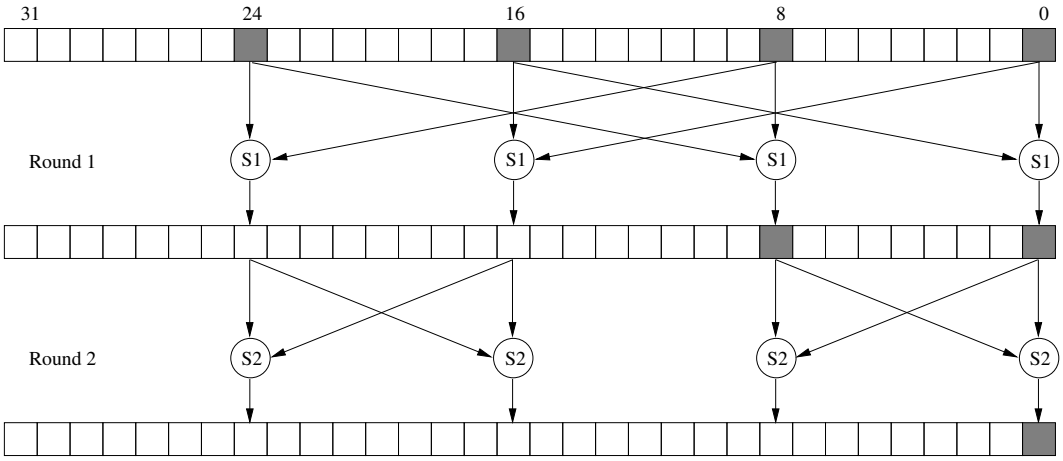


Figure 2.6: "Narrow pipe" in the first two rounds of COMP128's hash function

$RAND$, e.g., bytes 16 and 24 (see Figure 2.6), and wait for a collision at the output. Once a collision is detected, the two secret bytes of K_i involved, corresponding to the example above bytes 0 and 8, can be found by quick computer simulation. M.Briceno *et al.* demonstrated [BGW98b] that it takes about 165,000 authentication requests to the SIM card to extract the full session key K_c, which takes about 8 hours[8]. Surprisingly, they found out that 10 out of 64 bits of session key K_c were zero for all SIM cards they examined. This weakness has obviously been placed into COMP128 by intention of the designers.

[8]The bottleneck is the card reader!

3 DES Collision Attack

3.1 Introduction to DES

The Data Encryption Standard (DES) was developed in the mid 1970s by IBM. However, it is believed that the selection of the substitution boxes (S-Boxes) inside DES took place with input from the National Security Agency (NSA). DES became a public standard in the USA in 1977 by the National Bureau of Standards. Since then DES has been the most popular symmetric-key cipher in use world-wide. Even though a more secure successor of DES, the Advanced Encryption Standard (AES), has been chosen in 2001, DES is still widely used today. One example is the authentication of SmartCards with terminal devices (e.g., the German *Geldkarte* [Sel02]).

The DES maps 64 bits of plaintext to 64 bits of ciphertext using a 56 bit key.

$$DES : \{0,1\}^{56} \times \{0,1\}^{64} \rightarrow \{0,1\}^{64} \tag{3.1}$$

The input data is initially permuted and then processed in 16 rounds using the Feistel cipher. The Feistel cipher provides a bijective mapping: $G^{-1}(G(L,R)) = (L,R)$. It embeds an arbitrary function f_k, which does not need to be invertible (see Figure 3.1).

The function f_k, which is used in DES, is shown in Figure 3.2. It first expands the 32 input bits to 48 bits, which increases the dependency of the output bits on the input bits[1]. After the expansion, the corresponding 48 bit round key is x-ored. Round keys are derived from the original 56 bit key in a key scheduling algorithm either prior to DES encryption/decrpytion or on-the-fly prior to each round. The resulting 48 bits are then reduced back to 32 bits using eight non-linear S-Boxes. Finally, the 32 output bits of the S-Boxes are permuted. After five rounds, every ciphertext bit is a function of every plaintext bit and every key bit [Sch96]. Each S-Box substitutes a 6 bit input by a 4 bit output. Bits 5 and 0 of the 6 bit input

[1] diffusion

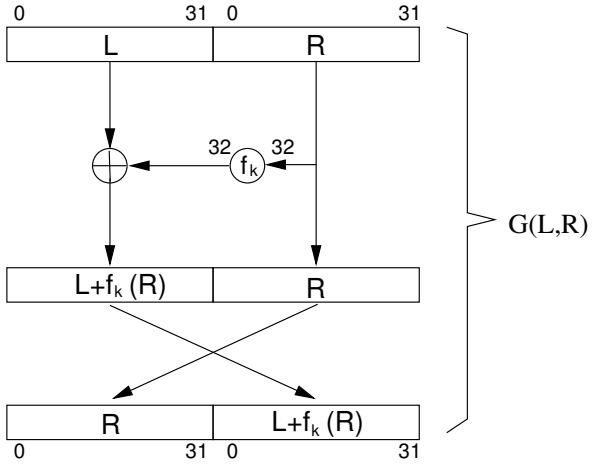


Figure 3.1: Feistel Cipher.

value encode the row and bits 4 to 1 encode the column information of the 4 bit output value within the S-Box. The eight S-Boxes are listed in Appendix A.1. Even though the content of the S-Boxes became public in 1977, the design criteria of the S-Boxes were not published until 1992 by Don Coppersmith [Cop94]:

1. Each S-Box has 6 input bits and 4 output bits (common register length of microchips in the mid 1970's).
2. No output bit should be close to a linear function of the input bits.
3. If the lowest and the highest bit of the input are fixed and the 4 middle bits are varied, each of the possible 4 bit output values is attained exactly once.
4. If two inputs to an S-Box differ in exactly one bit, their outputs must differ in at least two bits.
5. If two inputs to an S-Box differ in the two middle bits, their outputs must differ in at least two bits.
6. If two inputs to an S-Box differ in their first 2 bits and are identical in their last 2 bits, the two outputs must not be the same.
7. For any nonzero 6-bit difference between inputs, no more than 8 of the 32 pairs of inputs exhibiting that difference may result in the same output difference. [2]

[2] see the δ-tables in Appendix A.2

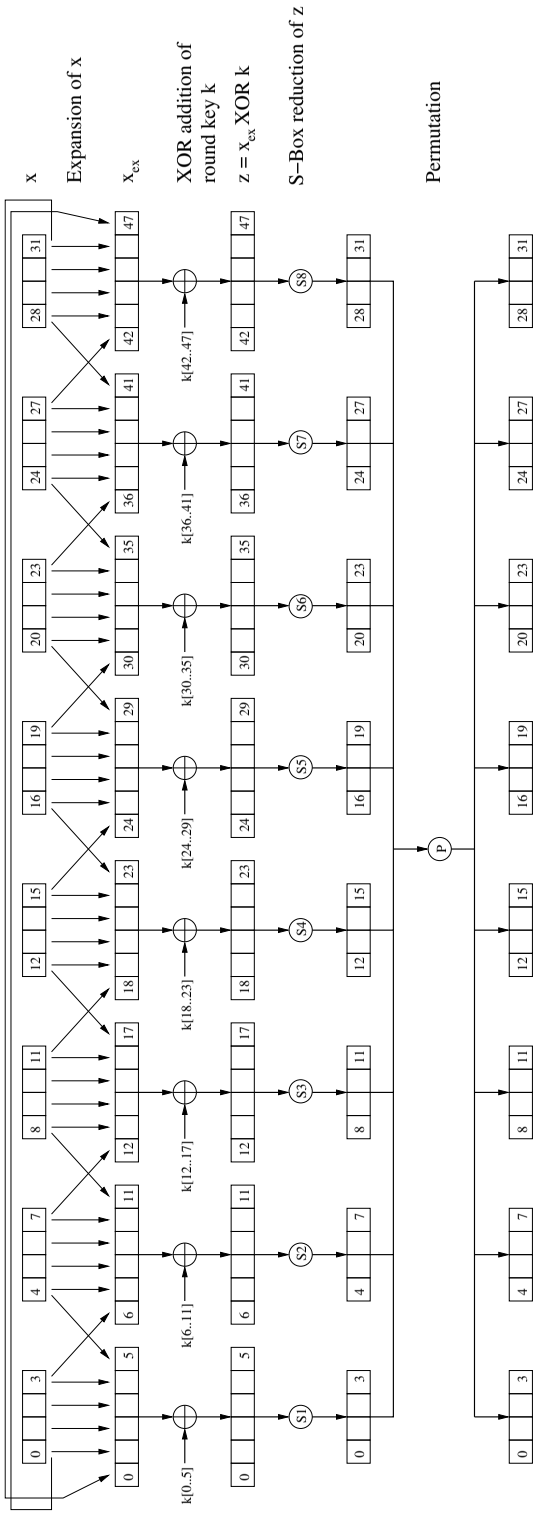


Figure 3.2: Detailed description of the non-linear round key function f_k used in DES.

3.2 Wiemers' collision attack against DES

The discussion of a collision attack against DES was originally initiated by Andreas Wiemers and Prof. Hans Dobbertin of the *Bundesamt für Sicherheit in der Informationstechnik* (BSI). The idea of the attack is as follows: an adversary performs a column jump within a chosen S-Box of round one and tries to compensate the change of the S-Box output by varying the corresponding bits of register L, which are x-ored to the observed S-Box output. As a reminder, the inputs to the Feistel cipher are two 32 bits wide registers L and R (see Figure 3.1). Register R enters the non-linear function f_k, which maps 32 input bits to 32 output bits. The output $f_k(R)$ is x-ored with register L. The Feistel cipher outputs the two 32 bits registers R and $L \oplus f_k(R)$. Any change of bits at the output of $f_k(R)$ can be compensated, if the corresponding bits of register L change as well. In this case, the output of the x-or addition does not change which can be understood as a collision. The non-linear function f_k in the succeeding Feistel round will then process the same input data $L \oplus f(R_k)$ even though the inputs R and L have changed. For example, if bit 10 of the output of $f_k(R)$ is inverted, bit 10 of register L must be inverted and the x-or sum $L \oplus R(f_k)$ will remain equal. Particular bits of a register can be simply inverted by x-oring an appropriate bit mask to this register. As in the example above, a bit mask with all bits zero except for bit 10 must be x-ored to register L in order to flip it.

Inside function f_k, eight S-Boxes map six input bits to four output bits. Due to the S-Box design criteria a change of a single S-Box input bit will at least alter two of the four output bits. If the output of an S-Box changes due to a change of an input bit, at least two of the corresponding four bits of register L must be inverted in order to cause a collision. This also narrows the number of possible bit combinations down to eleven: $0011, 0101, 0110, 0111, 1001, 1010, 1011, 1100, 1101, 1110, 1111$. A transition of the corresponding bits of register L which is equal to the actual S-Box output transition will cause an internal collision within the x-or operator of the Feistel cipher. If no countermeasures are provided in the implementation, this can be easily detected by correlating the power traces of the succeeding round. Once the adversary detects a collision for a particular bit combination in register L, he knows which bits have changed at the S-Box output. The S-Box input z can then be determined by simple analysis of the S-Box table, which finally reveals the corresponding six bit key $k = z \oplus x$.

In more detail, an adversary begins the attack by measuring the power trace

of function f_k in the second round. This power trace is going to be used as a reference trace, i.e., succeeding power traces of f_k of round 2 will be compared[3] to this trace. Next, the adversary performs a column jump within a chosen S-Box of round one by varying bits 2 and 3 of the corresponding S-Box input x within register R. It is not advantageous to alter Bits 0,1 and 4,5 of input x because these bits will also enter the adjacent S-Boxes due to the bit spreading in the expansion box. Then the outputs of the adjacent S-Boxes will change as well, which is not desired, because this would require the adversary to vary up to 8 or respectively 12 bits of register L in order to cause a collision. Therefore, only bits 2 and 3 will change the output state of a particular S-Box. This is shown in Figure 3.3.

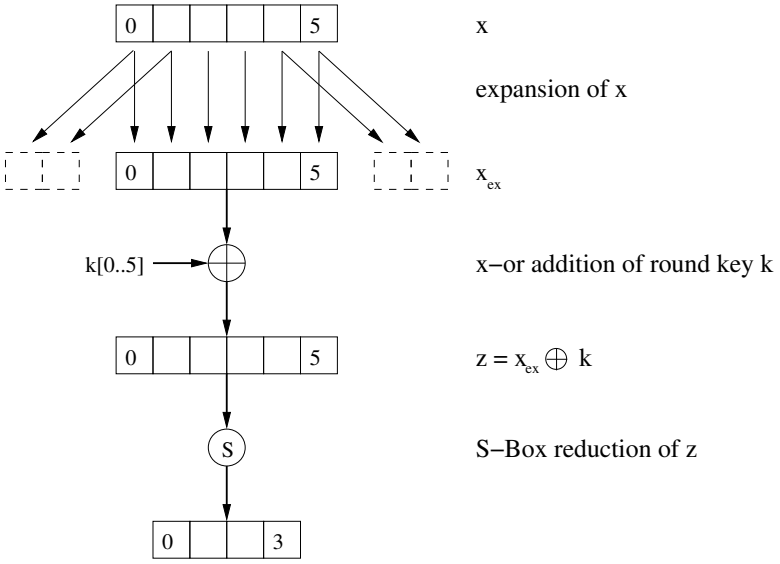


Figure 3.3: Influence of bit spreading in the expansion box

The column jump within the S-Box can also be viewed as an x-or addition of a differential δ to input x and, thus, input z of the S-Box[4]. Here, only three differentials $\delta_1 = 000100$, $\delta_2 = 001000$ and $\delta_3 = 001100$ are possible, which correspond to jumps by 2, 4 or 6 columns, respectively. In order to compensate the change of the S-Box output, i.e., output differential $\epsilon = S(z) \oplus S(z \oplus \delta)$, the adversary varies up to four bits of register L, which are x-ored to the output bits of the S-Box. Therefore 15 different bit combinations have to be examined by the adversary. For each bit combination, the adversary measures the power trace of

[3] by correlation

[4] The x-ored δ propagates through the key x-oring.

f_k in round two until the measured trace correlates noticeable high (depending on the target hardware) with the reference trace. This generally reveals a collision, i.e., it is assumed that the output differential ϵ is equivalent to the x-ored four bit combination of register L and thus the input to function f_k in round two is identical.

Once a collision has been detected for a particular input differential δ and an output differential ϵ, analysis of the correponding S-Box table reveals possible S-Box inputs z. If the S-Box input z is known, six key bits can be computed with the equation $k = z \oplus x$.

> **Example 3.1**: An adversary tries to determine the six key bits corresponding to S-Box 1 in the first round. The 32 bit input registers L and R of round one can be freely varied by the adversary. He/she encrypts an arbitrary plaintext (e.g., 64 zero bits), measures the power trace of f_k in round two and stores it as the reference trace. Since the DES key is secret, the inputs of the eight S-Boxes are unkown. However, the adversary is able to column jump within a particular S-Box of round one without affecting the adjacent S-Boxes. In this case, it is assumed that he/she x-ors input differential $\delta = 000100$ to input x of S-Box 1 in order to jump by two columns within a row. In Figure 3.4 all possible output differentials ϵ of S-Box 1 corresponding to this input differential are shown (in decimal). As stated in [MvOV97] the four output bits of S-Box 1 are permuted to bits 16,7,20 and 21 of the 32 bit output of function f_k. Therefore, the adversary needs to vary bits 16,7,20,21 of register L in order to compensate the output differential ϵ. He measures the power trace of f_k in round two for each bit combination and correlates it with the reference trace. In theory, a very high correlation will then reveal a collision, i.e., the four bit combination that was x-ored to register L by the adversary is equal to the output differential ϵ. Here, we assume that he begins with bit combination $0001, 0010, 0011, ...$ and finally detects a collision for $\epsilon = 0111 = 7|_{10}$.
>
> As shown in Figure 3.4 there exist six possible S-Box inputs $z_1, z_2, ..., z_6$, which correspond to $\epsilon = 7|_{10}$. Thus the adversary can compute six key candidates $k_i = z_i \oplus x$, with $i = 1, 2, ..., 6$. The six key bits are effectively reduced to $\log_2(6) \approx 2.6$ bits for a succeeding brute force attack.

As mentioned above, this example shows that not all 15 possible output dif-

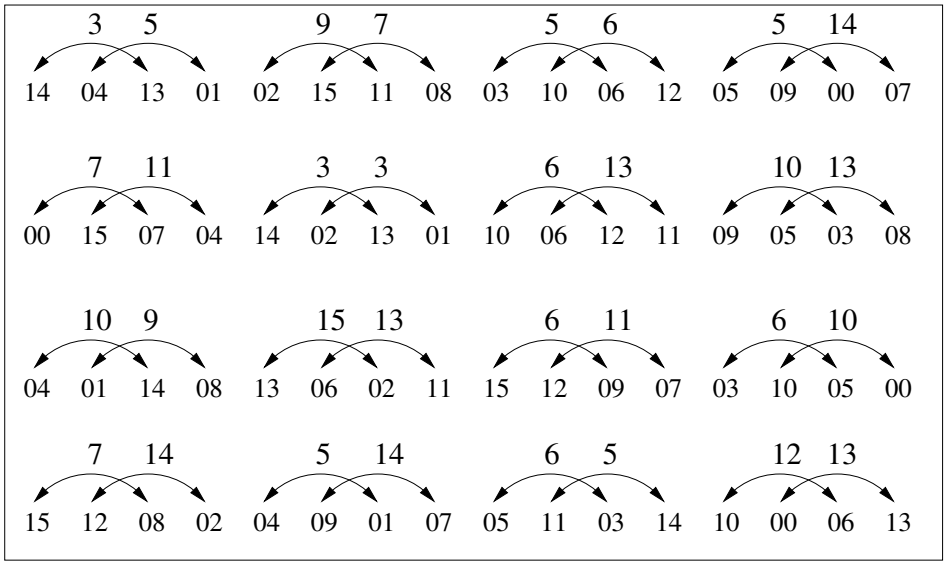


Figure 3.4: X-or output differentials ϵ of S-Box 1 (in decimal) corresponding to input differential $\delta = 000100$.

ferentials $\epsilon = 0001, ..., 1111$ occur for a particular S-Box and an input differential δ. Each value of ϵ corresponds to a particular number of S-Box inputs z_i with $i = 1, 2, ..., N$. In general N is an even number since every instance of an occuring output differential ϵ corresponds to a pair of S-Box inputs z_i and $z_{i+1} = z_i \oplus \delta$. Therefore there exist N key candidates $k_1, ..., k_N$. The less output differentials ϵ exist, the less bit combinations (i.e., power trace measurements) need to be examined by the adversary. However, this also implies that generally more S-Box inputs z_i and thus keys k_i correspond to an ϵ, which increases the costs of a succeeding brute force attack. An adversary, who targets on a particular S-Box, will typically choose a δ for which the least output differentials ϵ exist in order to minimize measurement costs. Tables 3.1 to 3.8 list all existing S-Box inputs z_i corresponding to a particular input differential δ and an output differential ϵ. In the example above, an adversary caused a collision in S-Box 1 with $\delta = 000100$ and $\epsilon = 7$. As shown in Table 3.1 six S-Box inputs $z_i \in \{10, 14, 1, 5, 33, 37\}$ are possible, which yield six six-bit key key candidates $k_i = x \oplus z_i$. These tables reveal that there generally exist between 9 and 11 values of ϵ for a particular S-Box/δ pair. Thus an adversary would typically choose a δ for a particular S-Box, which enables him to detect a collision with 9 measurements at most.

δ	ϵ	$z_1, z_2, \ldots$
$\delta = 000100$	3	0,4,9,13,11,15
	5	2,6,16,20,24,28,41,45,51,55
	6	18,22,17,21,48,52,56,60,49,53
	7	10,14,1,5,33,37
	9	8,12,34,38
	10	25,29,32,36,58,62
	11	3,7,50,54
	12	57,61
	13	19,23,27,31,42,46,59,63
	14	26,30,35,39,43,47
	15	40,44
$\delta = 001000$	3	18,26,17,25,19,27,23,31,38,46,55,63
	5	7,15,35,43,39,47,53,61
	6	4,12,16,24,20,28,50,58
	7	34,42,54,62
	9	6,14,32,40,37,45
	10	5,13
	11	2,10,22,30,33,41,51,59
	12	0,8,36,44,48,56,52,60
	13	3,11
	14	1,9
	15	21,29,49,57
$\delta = 001100$	3	16,28,20,24,36,40,49,61
	5	0,12,22,26,21,25
	6	7,11,32,44,51,63
	9	5,9,17,29,53,57
	10	34,46,48,60,52,56
	11	35,47,39,43
	12	2,14,50,62,37,41
	13	18,30,1,13,54,58
	14	6,10,3,15,19,31,23,27,38,42,33,45,55,59
	15	4,8

Table 3.1: Input values z of S-Box 1, with $\epsilon = S_1(z) \oplus S_1(z \oplus \delta)$.

δ	ϵ	$z_1, z_2, \ldots$
$\delta = 000100$	5	8,12,34,38,42,46
	7	0,4,1,5,9,13,32,36,40,44,33,37,41,45
	9	24,28,48,52,35,39
	10	18,22,26,30,3,7,19,23,51,55
	11	16,20,56,60
	12	11,15,27,31,58,62,49,53,59,63
	13	17,21,25,29,43,47
	14	50,54,57,61
	15	2,6,10,14
$\delta = 001000$	3	39,47,51,59
	5	16,24,55,63
	7	18,26,20,28,22,30,35,43
	9	0,8,7,15,19,27,54,62,53,61
	10	2,10,6,14,17,25,21,29,32,40,34,42,36,44,38,46
	11	4,12,50,58,49,57
	12	1,9,5,13,48,56
	14	52,60,33,41,37,45
	15	3,11,23,31
$\delta = 001100$	3	3,15,23,27
	5	2,14,6,10,7,11,19,31,52,56,54,58,49,61
	7	17,29,21,25,48,60,50,62,53,57
	9	33,45,37,41,55,59
	10	35,47
	11	1,13,5,9
	12	0,12,16,28
	13	18,30,22,26,32,44,36,40
	14	4,8,20,24,39,43
	15	34,46,38,42,51,63

Table 3.2: Input values z of S-Box 2, with $\epsilon = S_2(z) \oplus S_2(z \oplus \delta)$.

δ	ϵ	$z_1, z_2, \ldots$
$\delta = 000100$	3	0,4,25,29
	5	9,13
	6	10,14,19,23
	7	17,21
	9	8,12,24,28,32,36,48,52,57,61,59,63
	10	18,22,27,31,35,39,49,53
	11	40,44,56,60
	12	26,30,33,37,51,55
	13	16,20,1,5,50,54,58,62
	14	2,6,3,7,11,15,41,45,43,47
	15	34,38,42,46
$\delta = 001000$	3	2,10,3,11,7,15,19,27,35,43
	5	32,40,37,45
	6	4,12,5,13
	7	36,44,33,41,39,47
	9	18,26,34,42,38,46
	10	16,24,21,29,51,59
	11	6,14,50,58,54,62
	12	0,8,52,60,53,61
	14	20,28,1,9,17,25,48,56
	15	22,30,23,31,49,57,55,63
$\delta = 001100$	3	16,28,22,26,5,9,51,63
	5	0,12,2,14,18,30,23,27,48,60,53,57
	6	34,46,38,42,50,62,54,58,49,61,55,59
	7	20,24,52,56
	9	19,31,21,25,33,45,39,43
	11	1,13,37,41
	12	36,40
	13	6,10,3,15,7,11,17,29,35,47
	14	32,44
	15	4,8

Table 3.3: Input values z of S-Box 3, with $\epsilon = S_3(z) \oplus S_3(z \oplus \delta)$.

δ	ϵ	$z_1, z_2, \ldots$
$\delta = 000100$	3	26,30,27,31,32,36,33,37
	6	1,5,9,13,17,21,34,38,42,46,58,62
	7	18,22,41,45
	9	0,4,8,12,16,20,35,39,43,47,59,63
	11	19,23,40,44
	12	10,14,11,15,48,52,49,53
	13	3,7,56,60
	14	2,6,57,61
	15	24,28,25,29,50,54,51,55
$\delta = 001000$	3	26,30,27,31,32,36,33,37
	6	1,5,9,13,17,21,34,38,42,46,58,62
	7	18,22,41,45
	9	0,4,8,12,16,20,35,39,43,47,59,63
	11	19,23,40,44
	12	10,14,11,15,48,52,49,53
	13	3,7,56,60
	14	2,6,57,61
	15	24,28,25,29,50,54,51,55
$\delta = 001100$	3	20,24,21,25
	5	6,10,16,28,36,40,50,62
	6	23,27,52,56
	7	2,14,48,60,35,47,39,43
	9	22,26,53,57
	10	7,11,17,29,37,41,51,63
	11	3,15,34,46,38,42,49,61
	12	54,58,55,59
	13	18,30,1,13,5,9,32,44
	14	0,12,4,8,19,31,33,45

Table 3.4: Input values z of S-Box 4, with $\epsilon = S_4(z) \oplus S_4(z \oplus \delta)$.

δ	ϵ	$z_1, z_2, \ldots$
$\delta = 000100$	3	24,28,48,52,41,45,43,47
	5	32,36,42,46
	6	0,4,11,15,56,60,49,53,51,55
	7	3,7,33,37,59,63
	9	6,30,9,13,34,38
	10	18,22,17,21,19,23
	11	16,20,25,29
	12	8,12,10,14,1,5,50,54
	13	2,6,40,44,58,62
	15	27,31,35,39,57,61
$\delta = 001000$	3	38,46
	5	0,8,16,24,18,26,53,61
	6	2,10,22,30,17,25,36,44,35,43
	7	6,14,21,29
	9	19,27,48,56
	10	1,9,50,58,33,41,39,47,55,63
	11	54,62,51,59
	12	3,11,23,31,52,60,49,57
	13	20,28,7,15
	14	32,40,37,45
	15	4,12,5,13,34,42
$\delta = 001100$	3	4,8,1,13,23,27,32,44,49,61
	5	35,47
	6	16,28,5,9,19,31,38,42,54,58
	7	50,62
	9	0,12,33,45,39,43
	10	2,14,3,15,34,46,52,56,53,57
	11	6,10,7,11,36,40
	12	18,30,21,25,51,63
	13	17,29,37,41,55,59
	14	20,24
	15	22,26,48,60

Table 3.5: Input values z of S-Box 5, with $\epsilon = S_5(z) \oplus S_5(z \oplus \delta)$.

δ	ϵ	$z_1, z_2, \ldots$
$\delta = 000100$	3	16,20,25,29,27,31,48,52,
	6	0,4,32,36,33,37,41,45,
	9	18,22,11,15,51,55,
	10	10,14,50,54,56,60,49,53,
	11	24,28,17,21,34,38,42,46,58,62,
	12	26,30,
	13	3,7,59,63,
	14	2,6,1,5,9,13,40,44,57,61,
	15	8,12,19,23,35,39,43,47,
$\delta = 001000$	3	2,10,3,11,36,44
	5	0,8
	6	20,28,17,25,23,31,34,42,38,46,48,56,35,43,39,47
	7	6,14,7,15
	9	53,61
	10	18,26,19,27,55,63
	11	32,40
	12	4,12,54,62
	13	1,9,5,13,50,58,33,41,37,45,49,57
	14	16,24,21,29,51,59
	15	22,30,52,60
$\delta = 001100$	3	4,8,22,26,1,13,5,9,49,61,51,63
	5	16,28,17,29,23,27,32,44,52,56
	6	18,30,50,62
	7	54,58,53,57,55,59
	9	2,14,19,31,35,47,39,43
	10	0,12,3,15
	11	33,45,37,41
	12	48,60
	13	6,10,20,24,21,25,34,46,36,40,38,42
	14	7,11

Table 3.6: Input values z of S-Box 6, with $\epsilon = S_6(z) \oplus S_6(z \oplus \delta)$.

δ	ϵ	$z_1, z_2, \ldots$
$\delta = 000100$	3	24,28,11,15,35,39,43,47
	5	2,6,9,13
	6	0,4,1,5
	7	8,12,3,7,50,54,58,62
	9	27,31,34,38,56,60,49,53
	10	16,20,25,29,32,36,51,55
	11	18,22,26,30,17,21,40,44,33,37,41,45
	12	48,52
	13	10,14,42,46,57,61
	14	59,63
	15	19,23
$\delta = 001000$	3	6,14,38,46,53,61,55,63
	6	16,24,18,26,22,30
	7	34,42,33,41,37,45,49,57,51,59
	9	1,9,3,11
	10	4,12,5,13,23,31,48,56,50,58,54,62
	11	0,8,2,10
	12	17,25,19,27,36,44
	13	7,15,21,29,32,40
	15	20,28,52,60,35,43,39,47
$\delta = 001100$	3	23,27,48,60
	5	16,28,19,31
	6	2,14,17,29,32,44,52,56
	7	21,25,36,40
	9	51,63
	10	3,15,34,46,49,61
	12	0,12,20,24,1,13,33,45,35,47,37,41,39,43
	13	4,8,18,30,22,26,50,62,54,58,55,59
	14	6,10,7,11,38,42,53,57
	15	5,9

Table 3.7: Input values z of S-Box 7, with $\epsilon = S_7(z) \oplus S_7(z \oplus \delta)$.

δ	ϵ	$z_1, z_2, \ldots$
$\delta = 000100$	3	32,36
	5	0,4,57,61
	6	2,6,35,39,49,53
	7	18,22,26,30,3,7,11,15,40,44,43,47
	9	16,20,24,28,25,29
	10	17,21,34,38,48,52,56,60
	11	50,54,58,62
	12	1,5,27,31,33,37,41,45,51,55
	13	8,12,9,13
	14	10,14,19,23,42,46,59,63
$\delta = 001000$	3	4,12,38,46
	5	6,14,50,58,54,62
	6	33,41,37,45
	7	34,42
	9	18,26,22,30,23,31,51,59
	10	5,13,36,44,39,47
	11	0,8,1,9,19,27,35,43,55,63
	12	3,11,7,15,17,25,49,57
	13	2,10
	14	32,40
	15	16,24,20,28,21,29,48,56,52,60,53,61
$\delta = 001100$	3	2,14
	5	17,29,23,27,48,60,52,56,55,59
	6	0,12,16,28,20,24,1,13,21,25
	7	5,9,19,31,51,63
	9	32,44,34,46,49,61
	10	33,45,37,41,53,57
	11	6,10,3,15,7,11
	12	35,47
	13	36,40,38,42,39,43
	14	4,8,18,30,22,26,50,62,54,58

Table 3.8: Input values z of S-Box 8, with $\epsilon = S_8(z) \oplus S_8(z \oplus \delta)$.

The DES collision attack by Andreas Wiemers is a generic attack against Feistel ciphers. In theory it can be modified to attack any cryptographic algorithm based on the Feistel cipher. In the DES case, the attack is very powerful in practice. To our knowledge this is the first attack against DES combining internal collisions and side-channel analysis. Considering the high potential of this attack we asked ourselves whether the following attack is possible to create collisions in the function f_k with no regard to the embedding cipher structure. Here, a collision means finding a pair x_1, x_2 such that $f_k(x_1) = f_k(x_2)$. Such an attack is independent of the Feistel structure of DES. The deduction and optimization of this attack is presented in the remainder of this thesis. Finally, in the concluding chapter it is compared to the collision attack of Andreas Wiemers. Even though Wiemers' attack is more powerful in pratice, we do believe that our modification is of theoretical interest and provides further insight into the non-linear f_k-function of DES.

3.3 Collisions in a Single S-Box

In cryptography a collision is generally defined for surjective mappings of a particular function, e.g., a hash function $h(M)$ which maps an input M of arbitrary length to a hash value of fixed length. If a collision occurs, different input values map to an equal output value:

$$h(M_1) = h(M_2) = ... = h(M_n), \qquad M_i \text{ pairwise different, } n \geq 2 \tag{3.2}$$

The eight S-Box mappings $2^6 \rightarrow 2^4$ of DES are surjective. Moreover, the mappings are uniformly distributed, which means that always four different inputs map to one specific output.

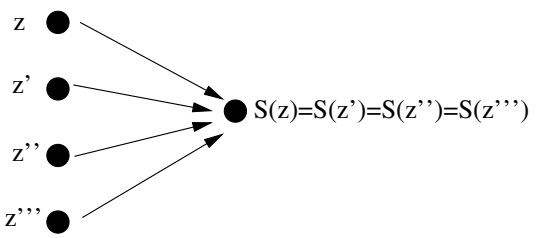


Figure 3.5: Surjective input-output mapping of the S-Boxes in DES.

Regarding the S-Boxes of DES, for each 6 bit input z exist three further input values z', z'' and z''', which will yield the same S-Box ouput:

$$S(z) = S(z') = S(z'') = S(z'''), \qquad z \neq z' \neq z'' \neq z''' \tag{3.3}$$

The values z', z'' and z''' can also be substituted by x-or sums of input z and specific 6 bit differences δ_1, δ_2 and δ_3:

$$S(z) = S(\underbrace{z \oplus \delta_1}_{z'}) = S(\underbrace{z \oplus \delta_2}_{z''}) = S(\underbrace{z \oplus \delta_3}_{z'''}), \qquad \delta_1 \neq \delta_2 \neq \delta_3 \neq 0 \tag{3.4}$$

Example 3.1: If the first S-Box is examined and $z = 000000$:

$$\begin{aligned} S_1(000000) &= S_1(000000 \oplus 001001 = 001001) \\ &= S_1(000000 \oplus 100100 = 100100) \\ &= S_1(000000 \oplus 110111 = 110111) = 14 \end{aligned}$$

The x-or addition of the three differentials $\delta_1 = 001001$, $\delta_2 = 100100$ and $\delta_3 = 110111$ to input value $z = 000000$ is also shown in Figure 3.6, in which each arrow symbolizes an x-or addition of δ_i.

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Figure 3.6: Collisions $S(z) = S(z \oplus \delta_i)$ inside S-Box 1 for $z = 000000$.

The propagation of a collision inside DES is shown in Figure 3.7. It is assumed that only those bits of register R are changed which will enter the targeted S-Boxes. The remaining bits of register R and all 32 bits of register L are fixed. The specific bits of register R are varied until a collision occurs in the compromised S-Boxes, which means that the output of $f_{k,n}$ remains unchanged. Function $f_{k,n+1}$ of the next round $n+1$ will then process the same input data. The bold arrows emphasize the propagation path of a collision taking place in the S-Boxes of round n. Due to the Feistel cipher a collision is only detectable in the succeeding round. As shown in Figure 3.7 register R in round n is fed into function $f_{k,n}$, but it is

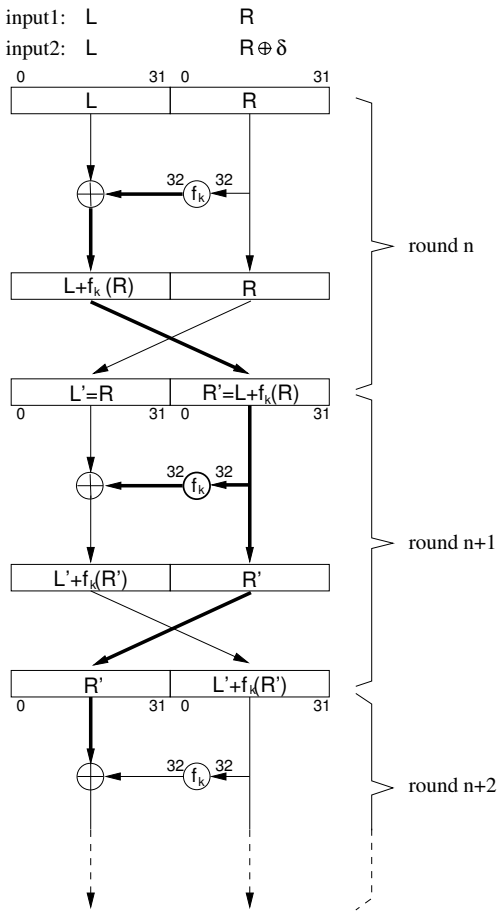


Figure 3.7: Propagation of a collision in the S-Boxes.

also directly connected to the output of round n. Hence, the propagation of a collision in round n will end in round $n + 2$. Thus, a collision is not detectable at the output of DES as in practice $n = 1$ (in other ciphers such as COMP128 internal S-Box collisions can be detected at the output [Zen99]). However, side channel analysis can be used to detect internal collision. In this thesis, power traces of the succeeding round were measured and correlated in order to detect collisions. This is explained in more detail in Chapter 5.

First we restrict ourselves to the study of a single S-Box while ignoring the expansion function. As stated above, there exist three specific deltas δ_1, δ_2 and δ_3 for each of the 64 possible S-Box input values z causing the collisions $S(z) = S(z \oplus \delta_1)$, $S(z) = S(z \oplus \delta_2)$ and $S(z) = S(z \oplus \delta_3)$. In Appendix A.2 eight tables list all occurring differentials δ and their corresponding values of z, which fulfill the condition $S(z) = S(z \oplus \delta)$ for each S-Box. The input values z in this table

always occur in pairs of z and $z' = z \oplus \delta$, because both z and z' fulfill the condition $S(z) = S(z \oplus \delta) \Leftrightarrow S(z') = S(z' \oplus \delta)$. The minimum number of z values corresponding to a δ that was found is zero, which means that certain differentials δ do not exist (for example $\delta = 011010$, only causes a jump within an S-Box row, and no S-Box ouput value occurs twice in a row). The maximum number of values of z corresponding to a δ that was found is 16. This number is also confirmed by the S-Box design criteria in [Cop94].

Example 3.2: An adversary wants to cause a collision in S-Box 1. He chooses $\delta = 000111$ and looks at the δ-table of S-Box 1 in Appendix A.2 to find out which input values z can cause a collision $S(z) = S(z \oplus \delta)$ with this particular δ.

δ	#z	(z_1,$z_1 \oplus \delta$), ...
...	...	...
000111	2	((010011(9,1),010100(10,0))
...	...	...

Table 3.9: Input values z of S-Box 1, with $S_1(z) = S_1(z \oplus \delta)$.

He finds out that there exist two S-Box input values $z_1 = 010011$ and $z_2 = z_1 \oplus \delta = 010100$, which cause a collision $S_1(z) = S_1(z \oplus \delta)$. As shown in Figure 3.3 the input value z is the x-or sum of the expanded input x_{ex} and the round key k:

$$z = x_{ex} \oplus k \Leftrightarrow k = z \oplus x_{ex}$$

The attacker randomly varies x (and thus effectively x_{ex}) until he detects a collision $S_1(x_{ex} \oplus k) = S_1(x_{ex} \oplus k \oplus \delta)$. It is assumed that a collision is detected at input $x_{ex} = 101000$. Then two possible key candidates k_1 and k_2 can be determined:

$$\begin{aligned} k_1 &= z_1 \oplus x_{ex} = 010011 \oplus 101000 = 111011 \\ k_2 &= z_2 \oplus x_{ex} = 010100 \oplus 101000 = 111100 \end{aligned}$$

However, the attack of the last example will not be successful: the bit spreading of the expansion box makes it impossible to control all six input bits of an S-Box without altering the input bits of the adjacent S-Boxes. The two most and least

significant bits of the x-or sum ($x_{ex} \oplus \delta$) will also enter S-Boxes 8 and 2. Even if a collision takes place in S-Box 1, the inputs (and outputs) of S-Boxes 8 and 2 will change unless the two most and least significant bits of the x-ored δ are zero (see Figure 3.8). Hence, if the two most and least significant bits of differential δ are zero, the input of the adjacent S-Boxes 8 and 2 will not change after x-or addition of δ. It is assumed that the adversary uses power analysis of the succeeding round to detect collisions of the overall 32 bit output of the S-Boxes. Thus it is not possible to observe a single S-Box collision while ignoring the outputs of the other S-Boxes. In order not to alter the inputs of the adjacent S-Boxes, δ must comply with the following bit mask:

$$\delta = 00x_1x_200 \qquad x_i \in \{0,1\}$$

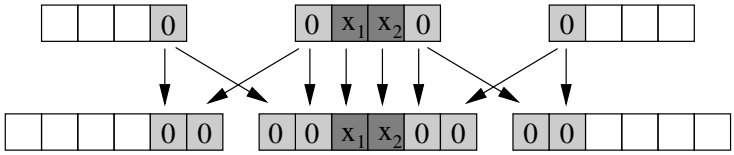


Figure 3.8: Required Bit Mask of δ for a Single S-Box Collision.

However, analysis of the δ-tables in Appendix A.2 reveals that no such x-or differentials δ exist for any S-Box which also follows directly from the S-Box design criterion no. 5 on page 20. Therefore, collisions in single S-Boxes are not possible.

3.4 Collisions in Two S-Boxes

In the last section it was shown that it is not possible to cause a single S-Box collision $S(z) = S(z \oplus \delta)$ without altering the inputs of the adjacent S-Boxes. As next, an adversary might try to cause two collisions simultaneously within a pair of adjacent S-Boxes by concatenation of two δ-values of each S-Box: $\Delta = \delta_1|\delta_2$. The resulting Δ is of length 12 bits. In order not to alter the inputs of the adjacent S-Boxes, the two most and least significant bits of Δ must be zero[5]:

$$\Delta[0] = \Delta[1] = \Delta[10] = \Delta[11] = 0$$

[5] $\Delta[i]$ denotes bit i of Δ

In order to propagate through the expansion box, Δ must fulfil the condition:

$$\Delta[4] = \Delta[6], \Delta[5] = \Delta[7]$$

Therefore Δ must comply with the following bit mask:

$$\Delta = \delta_1|\delta_2 = 00x_1x_2vwvwx_3x_400 \qquad x_i, v, w \in \{0,1\}$$

The bit spreading of the concatenated 12 bit Δ and the implied Δ-bit mask is also shown in Figure 3.9. However, from the design criteria no. 5 and no.6 stated on page 20 we know that no collision will occur in a single S-Box for differentials $\delta = 0abcd0$ and differentials $\delta = 11ef00$ with $a, b, c, d, e, f \in \{0,1\}$. As a conclusion, no concatenated differential $\Delta = \delta_1|\delta_2$ complying with the bitmasks $\Delta_1, \ldots, \Delta_4$ can cause a simultaneous collision in an S-Box pair:

$$\begin{aligned}
\Delta_1 &= 00x_1x_200|00x_3x_400 \Rightarrow \text{no collision due to } \delta_1 \\
\Delta_2 &= 00x_1x_201|01x_3x_400 \Rightarrow \text{no collision due to } \delta_2 \\
\Delta_3 &= 00x_1x_210|10x_3x_400 \Rightarrow \text{no collision due to } \delta_1 \\
\Delta_4 &= 00x_1x_211|11x_3x_400 \Rightarrow \text{no collision due to } \delta_2
\end{aligned}$$

As a result, it is not possible to cause a collision in an S-Box pair while preserving the inputs of the two neighboring S-Boxes.

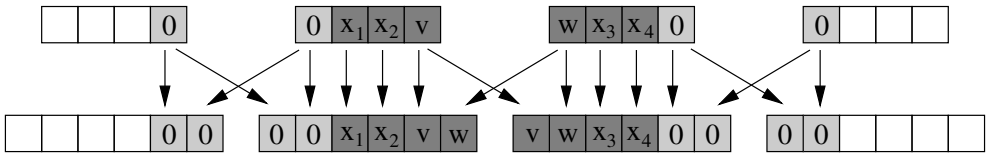


Figure 3.9: Required S-Box pair Δ bit mask.

3.5 Collisions in Three S-Boxes

It has been shown that collisions are neither possible in single S-Boxes nor simultaneously in S-Box pairs. However, an adversary might try to cause three simultaneous collisions within three adjacent S-Boxes. Three x-or differentials δ_1, δ_2 and δ_3 (one for each S-Box) are concatenated to $\Delta = \delta_1|\delta_2|\delta_3$. The resulting Δ

is of 18 bits length. In order not to alter the inputs of the two adjacent S-Boxes, the two most and least significant bits of Δ must be zero:

$$\Delta[0] = \Delta[1] = \Delta[16] = \Delta[17] = 0 \tag{3.5}$$

In order to propagate through the expansion box, Δ must fulfil the condition:

$$\Delta[4] = \Delta[6], \Delta[5] = \Delta[7], \Delta[10] = \Delta[12], \Delta[11] = \Delta[13] \tag{3.6}$$

Therefore Δ must comply with the following bit mask:

$$\Delta = \delta_1|\delta_2|\delta_3 = 00x_1x_2vwvwx_3x_4yzyzx_5x_600 \qquad x_i, v, w, y, z \in \{0, 1\} \tag{3.7}$$

The bit spreading of differential Δ caused by the expansion box and the applied bit mask is also shown in Figure 3.10.

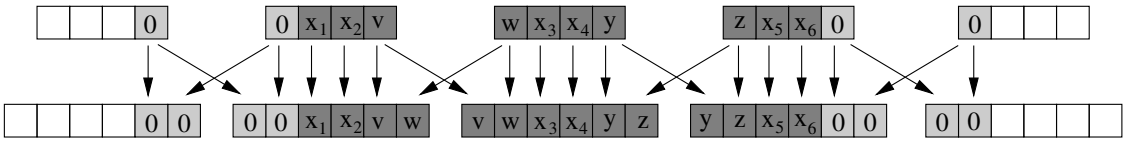


Figure 3.10: Required S-Box triple Δ Bit Mask.

Analysis of the δ-tables in Appendix A.2 shows that there exist many 18 bit x-or differentials $\Delta = \delta_1|\delta_2|\delta_3$ for all eight S-Box triples, which fulfils the bit mask $00x_1x_2vwvwx_3x_4yzyzx_5x_600$. The resulting differentials Δ are presented in Appendix A.3. Therefore, it is possible to cause collisions in an S-Box triple.

In order to cause a collision, an adversary randomly generates a 14 bit input x. These 14 bits of x correspond to 14 bits of the 32 bit register R entering the appropriate S-Box triple. Register L and the remaining 18 bits of R are not varied. Within function f_k the 14 bits of x are expanded to x_{ex} of length 18 bits. Then the x-or sum $x_{ex} \oplus k$ enters the three S-Boxes, whereas k denotes the 18 bits of the round key, which correspond to the S-Box triple. The adversary then sets the input to $x_{ex} \oplus \Delta$ and determines whether the resulting outputs $f_k(x_{ex})$ and $f_k(x_{ex} \oplus \Delta)$ coincide or not[6]. If the outputs $f_k(x_{ex})$ and $f_k(x_{ex} \oplus \Delta)$ are equal[7], a collision has occured. The possible S-Box inputs can then be found in the corresponding δ-tables of δ_1, δ_2 and δ_3 in Appendix A.2. If Z_1 is the set of

[6] $f_k(x_{ex})$ describes the 32 bit output of function f_k with x_{ex} being the expanded 18 bits of x corresponding to the S-Box triple.

[7] It is not required to know the particular values of $f(x_{ex})$ and $f_k(x_{ex} \oplus \Delta)$

all possible S-Box inputs corresponding to δ_1, Z_2 is the set of all S-Box inputs corresponding to δ_2 and Z_3 is the set of all S-Box inputs corresponding to δ_3 then the set of all possible input combinations is defined as

$$Z = Z_1 \times Z_2 \times Z_3 \tag{3.8}$$

Each element $z \in Z$ is a concatenated 18 bit input value to the S-Box triple. The set of all possible key candidates is then defined as

$$K = \{x_{ex} \oplus z | z \in Z\} \tag{3.9}$$

Therefore the order[8] of set Z and set K is equal

$$|Z| = |K| \tag{3.10}$$

> **Example 3.3**: An adversary intends to cause a simultaneous collision in the S-Box triple 1,2 and 3 in order to gain detailed information about the underlying 18 key bits. He chooses $\Delta = \delta_1|\delta_2|\delta_3 = 000011\,110010\,101100 = \Delta_3$ (see Appendix A.3), which is a concatenation of $\delta_1 = 000011$, $\delta_2 = 110010$ and $\delta_3 = 101100$. In the case of Δ_3 there exist 14 possible inputs to S-Box 1, 8 possible inputs to S-Box 2 and 10 possible inputs to S-Box 3 corresponding to δ_1, δ_2 and δ_3 respectively. Therefore there exist $14 \times 8 \times 10 = 1120$ possible key candidates $k = x_{ex} \oplus z$ for the 18 secret key bits. The adversary randomly generates x (14 bits), which is expanded to x_{ex} (18 bits) and measures $f(x_{ex})$ and $f(x_{ex} \oplus \Delta_3)$. Once a collision $f(x_{ex}) = f(x_{ex} \oplus \Delta_3)$ is detected for a particular value x_{ex}, the information content of the 18 key bits is reduced to $log_2(1120) \approx 10.13$ bits and the overall cost of a brute force attack decreases from 2^{56} to approximately 2^{48} steps.

It has been shown that for each 18 bit x-or input differential Δ of an S-Box triple, there exists a collision set Z of 18 bit S-Box triple input values. As next, it has to be determined, which collisions $z \in Z$ can occur for a particular key k. (i.e., the influence of k on possible collisions $z \in Z$). If it is assumed that a particular $k = x_{ex} \oplus z$ is fixed, the expanded input value x_{ex} can be expressed as $x_{ex} = k \oplus z$. However, x_{ex} is only valid, if it fulfils the expansion condition:

$$x_{ex}[4] = x_{ex}[6], x_{ex}[5] = x_{ex}[7], x_{ex}[10] = x_{ex}[12], x_{ex}[11] = x_{ex}[13] \tag{3.11}$$

[8] i.e., number of elements

Thus x_{ex} must comply with the following bitmask:

$$x_{ex} = x_1x_2x_3x_4vwvwx_5x_6yzyzx_7x_8x_9x_{10} \qquad x_i, v, w, y, z \in \{0, 1\} \tag{3.12}$$

Therefore, for a particular key k, only those inputs $z \in Z$ can cause a collision, for which $x_{ex} = k \oplus z$ fulfils the expansion condition. The set of valid S-Box triple inputs z causing a collision for a given key k is generally a subset of set Z:

$$Z_k \subseteq Z \tag{3.13}$$

This subset Z_k depends on eight bits of key k: $k[4], k[5], k[6], k[7]$ and $k[10], k[11], k[12], k[13]$, because only these bits of k will affect the expansion condition of x_{ex} stated above.

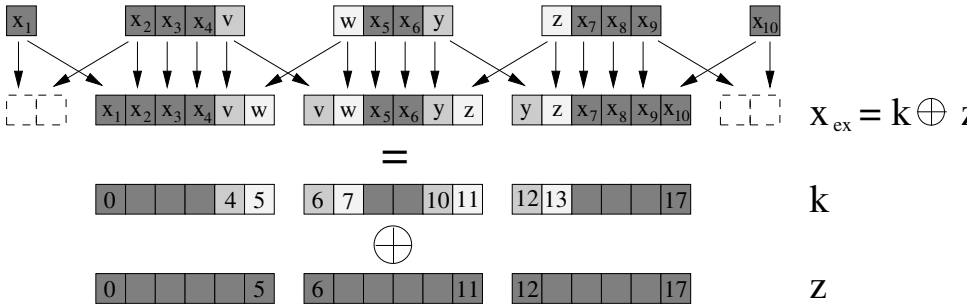


Figure 3.11: Influence of eight bits of k on the validity of x_{ex}.

Thus any 18 bit key k can be classified into one of 256 possible key sets K_i depending on bits 4–7 and 10–13 of k:

$$k \in K_i, 0 \leq i \leq 255 \tag{3.14}$$

The order of a key set K_i is

$$|K_i| = \frac{\#possible\ keys}{\#possible\ key\ sets\ K_i} = \frac{2^{18}}{256} = 2^{10} \tag{3.15}$$

For example, key set K_0 is the set of all keys k, for which bits 4-7 and 10-13 are zero:

$$K_0 = \{000000000000000000, 000000000000000001, ..., 111100001100001111\}$$

Example 3.4: An adversary tries to cause a collision in the S-Box triple 1,2,3 using the input differential Δ_1. Table 3.10 lists an excerpt

of the number of valid inputs $z \in Z$ (i.e., the number of possible collisions) to S-Box triple 1,2,3 for each key set K_i and for each Δ. In general the key k is secret, i.e., unknown to the adversary. Here, it is assumed that the 18 secret key bits are $k = 101100001000011111$ in order to computate the probability of a collision. The key k is thus an element of key set K_1.

key set K_i (binary)	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	...	Δ_{47}	Δ_{48}
xxxx0000xx0000xxxx	0	84	28	56	0	112	...	0	0
xxxx0000xx0001xxxx	**56**	0	112	0	84	28	...	0	0
xxxx0000xx0010xxxx	0	84	28	56	0	112	...	0	0
xxxx0000xx0011xxxx	56	0	112	0	84	28	...	0	0
xxxx0000xx0100xxxx	56	0	112	0	84	28	...	0	0
xxxx0000xx0101xxxx	0	84	28	56	0	112	...	0	0
xxxx0000xx0110xxxx	56	0	112	0	84	28	...	0	0
xxxx0000xx0111xxxx	0	84	28	56	0	112	...	0	0
...	...	...	...	...	...	...	...	...	
xxxx1111xx1011xxxx	56	0	112	0	84	28	...	0	0
xxxx1111xx1100xxxx	56	0	112	0	84	28	...	0	0
xxxx1111xx1101xxxx	0	84	28	56	0	112	...	0	0
xxxx1111xx1110xxxx	56	0	112	0	84	28	...	0	0
xxxx1111xx1111xxxx	0	84	28	56	0	112	...	0	0

Table 3.10: S-Box triple 1,2,3: number of collisions for a particular key set and a Δ.

In Appendix A.3 it is stated that the order of the collision set Z corresponding to S-Box triple 1,2,3 and Δ_1 is $|Z| = 448$. As shown in Table 3.10, there exists a subset $Z_k \subseteq Z$ of 56 inputs for key $k = 101100001000011111$ ($k \in K_1$), which will cause a collision for this key. The attacker randomly generates the 14 bit input x until he detects a collision $f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta_1)$. The probability that a collision occurs is

$$P(f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta_1) | k \in K_1) = \frac{56}{2^{14}} \approx 0.00342$$

Each collision test $f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta_1)$ requires two power trace measurements[9]. The average number of required power trace measure-

[9] Averaging of power traces in order to reduce noise is not taken into account here.

ments (i.e., the number of plain text encryptions) is given as

$$\overline{\#M} = 2 \cdot P(f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta_1) | k \in K_1)^{-1} = 2 \cdot \frac{2^{14}}{56} \approx 585$$

In average, the adversary needs to check $\lceil \frac{585}{2} \rceil \approx 293$ collision tests $f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta_1)$ until he detects a collision. As Table 3.10 reveals, once a collision occurs for a randomly generated x_{ex}, there exist 56 possible key candidates
$k = x_{ex} \oplus z$, $(z \in Z_k)$ for the 18 secret key bits. Thus, the information content of the 18 key bits is reduced to $\log_2(56) \approx 5.81$ bits and the overall cost of a brute force attack decreases from 2^{56} to about $2^{43.81}$ steps.

Example 3.5: An adversary tries to cause a collision in S-Box triple 1,2,3 using the input differential Δ_1. This time it is assumed that the 18 secret key bits are $k = 101100001000001111$. The key k is an element of K_0. As shown in Table 3.10 for key $k = 101100001000001111$ $(k \in K_0)$ do not exist any valid inputs z, which will cause a collision. Hence the probability of a collision is:

$$P(f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta_1) | k \in K_0) = \frac{0}{2^{14}} = 0$$

Therefore an adversary will not be able to detect any collisions using Δ_1. The attack will not be successful.

The last example shows that there exist input differentials Δ, which will not cause collisions for certain keys. However, there also exist values of Δ for which collisions will occur for any key (i.e., over all 256 key sets K_i). In the following these differentials will be referred to as *collision resistent*. For example, 10 out of 48 possible differentials Δ of S-Box triple 1,2,3 will cause collisions over all 256 key sets: Δ_3,Δ_6,Δ_9, Δ_{12},Δ_{15},Δ_{18}, Δ_{27},Δ_{30},Δ_{39} and Δ_{42}. All collision resistent differentials are listed in Tables 3.11 - 3.18 at the end of this chapter.

In general, no details of the key k are known. Therefore an adversary chooses a Δ, for which collisions will occur for any key. The total probability of a collision for a particular Δ over all 256 key sets is

$$P(f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta)) = \sum_{i=0}^{255} P(f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta) | k \in K_i) \cdot P(k \in K_i) \tag{3.16}$$

As stated above, there exists one collision subset $Z_k \subseteq Z$ for each of the 256 key sets K_i and thus for every $k \in K_i$. Moreover, for each $z \in Z$ there exist exactly 16 different key sets K_i, which comply with the expansion bit mask:

$$x_{ex} = z \oplus k = x_1x_2x_3x_4vwvwx_5x_6yzyzx_7x_8x_9x_{10}, k \in K_i$$

For that reason, the total probability of a collision can also be expressed as

$$\begin{aligned} P(f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta)) &= P(k \in K_i) \cdot \frac{16 \cdot |Z|}{2^{14}} && (3.17) \\ &= \frac{1}{256} \cdot \frac{16 \cdot |Z|}{2^{14}} = \frac{|Z|}{2^{18}} && (3.18) \\ & && (3.19) \end{aligned}$$

The average number of required measurements over all 256 key classes therefore is

$$\overline{\#M} = 2 \cdot \frac{1}{256} \cdot \sum_{i=0}^{255} \frac{1}{P(f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta) | k \in K_i)} \quad (3.20)$$

Example 3.6: An adversary tries to cause a collision in S-Box triple 1,2,3 using the input differential Δ_3, which will cause collisions for any key over all 256 key sets. The probability that key k is an element of key set K_i is $P(k \in K_i) = \frac{1}{256}$. The total probability of a collision is

$$\begin{aligned} P(f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta_3)) &= \sum_{i=0}^{255} P(f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta_3) | k_i) \cdot P(k \in K_i) \\ &= \frac{1}{256} \cdot \sum_{i=0}^{255} P(f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta_3) | k \in K_i) \\ &= \frac{1}{256} \cdot \frac{1}{2^{14}} \cdot (28 + 112 + \ldots + 28) \approx 0.004272 \end{aligned}$$

As shown above the total probability can also be determined directly by

$$P(f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta_3)) = \frac{|Z|}{2^{18}} = \frac{1120}{2^{18}} \approx 0.004272$$

The average number of required measurements until a collision occurs

is

$$\begin{aligned}\overline{\#M} &= 2 \cdot \frac{1}{256} \cdot \sum_{i=0}^{255} \frac{1}{P(f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta_3) | k \in K_i))} \\ &= 2 \cdot \frac{1}{256} \cdot 2^{14} \cdot (\frac{1}{28} + \frac{1}{112} + \ldots + \frac{1}{28}) \approx 731\end{aligned}$$

In average, the adversary needs to check $\lceil \frac{731}{2} \rceil \approx 366$ collision tests $f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta_3)$ until he detects a collision.

The total probability that a collision occurs for a particular Δ is proportional to the order $|Z|$ of the collision set Z corresponding to this Δ. In general, choosing a Δ with a maximum total probability (i.e., maximum order $|Z|$) will minimize the average measurement costs $\overline{\#M}$. However, the costs of a succeeding brute force attack are maximized since the number of key candidates k is $|Z|$. Choosing a Δ with a minimum total probability (i.e., minimum order $|Z|$) will maximize the average measurement costs $\overline{\#M}$. On the other hand, the costs of a succeeding brute force attack are minimized since the number of key candidates k is $|Z|$.

In Tables 3.11 - 3.18 for each S-Box triple all differentials Δ, which will cause collisions over all 256 key sets K_i , are listed. Moreover, the order of the corresponding collision set Z_Δ, the total probability of a collision and the average number of required measurements are listed. The least measurement costs can be achieved by using Δ_3 of S-Box triple 3,4,5. Only an average of 512 measurements is needed to detect a collision. The number of key candidates after a collision occurs is 1024 with this setup. The costs of a succeeding brute force attack can be minimized with Δ_{12} of S-Box triple 6,7,8. An average of 12288 measurements are needed to detect a collision. However, the number of key candidates after a collision occurs is only 48 with this setup.

Δ	$\lvert Z\rvert$	$P(collision) = \lvert Z\rvert \cdot 2^{-18}$	$\overline{\#M}$
$\Delta_3 = 000011110010101100$	1120	0.00427	731
$\Delta_6 = 000011110110101100$	1120	0.00427	731
$\Delta_9 = 000011111010101100$	560	0.00214	1792
$\Delta_{12} = 000011111110101100$	560	0.00214	1792
$\Delta_{15} = 000111110010101100$	160	0.00061	5120
$\Delta_{18} = 000111110110101100$	160	0.00061	5120
$\Delta_{27} = 001011110010101100$	160	0.00061	5120
$\Delta_{30} = 001011110110101100$	160	0.00061	5120
$\Delta_{39} = 001111110010101100$	160	0.00061	5120
$\Delta_{42} = 001111110110101100$	160	0.00061	5120

Table 3.11: S-Box triple 1,2,3: number of key candidates, total probability and average number of measurements for each collision resistent Δ.

Δ	$\lvert Z\rvert$	$P(collision) = \lvert Z\rvert \cdot 2^{-18}$	$\overline{\#M}$
$\Delta_3 = 000011110110101000$	256	0.00098	2048
$\Delta_7 = 000011111110101000$	512	0.00195	1365
$\Delta_9 = 000111110010101000$	128	0.00049	4096
$\Delta_{11} = 000111110110101000$	256	0.00098	2048
$\Delta_{13} = 000111111010101000$	384	0.00146	1365
$\Delta_{14} = 000111111010101000$	192	0.00073	3072
$\Delta_{15} = 000111111110101000$	512	0.00195	1024
$\Delta_{16} = 000111111110101100$	256	0.00098	2731
$\Delta_{17} = 001011110010101000$	192	0.00073	3072
$\Delta_{19} = 001011110110101000$	384	0.00146	1365
$\Delta_{21} = 001011111010101000$	576	0.00220	1024
$\Delta_{22} = 001011111010101100$	288	0.00061	2304
$\Delta_{23} = 001011111110101000$	768	0.00293	702
$\Delta_{24} = 001011111110101100$	384	0.00146	1997

Table 3.12: S-Box triple 2,3,4: number of key candidates, total probability and average number of measurements for each collision resistent Δ.

Δ	$\lvert Z\rvert$	$P(collision) = \lvert Z\rvert \cdot 2^{-18}$	$\overline{\#M}$
$\Delta_1 = 000011110010100100$	768	0.00293	683
$\Delta_2 = 000011110010101000$	256	0.00098	2048
$\Delta_3 = 000011110010101100$	1024	0.00391	512
$\Delta_4 = 000111110010100100$	192	0.00073	2731
$\Delta_5 = 000111110010101000$	64	0.00024	8192
$\Delta_6 = 000111110010101100$	256	0.00098	2048
$\Delta_7 = 001011110010100100$	192	0.00073	2731
$\Delta_8 = 001011110010101000$	64	0.00024	8192
$\Delta_9 = 001011110010101100$	256	0.00098	2048
$\Delta_{10} = 001111110010100100$	384	0.00146	1365
$\Delta_{11} = 001111110010101000$	128	0.00049	4096
$\Delta_{12} = 001111110010101100$	512	0.00195	1024

Table 3.13: S-Box triple 3,4,5: number of key candidates, total probability and average number of measurements for each collision resistent Δ.

Δ	$\lvert Z\rvert$	$P(collision) = \lvert Z\rvert \cdot 2^{-18}$	$\overline{\#M}$
$\Delta_1 = 000011110010100100$	320	0.00122	2560
$\Delta_2 = 000011110010101000$	192	0.00073	3072
$\Delta_{10} = 000111110010100100$	160	0.00061	5120
$\Delta_{11} = 000111110010101000$	96	0.00037	6144

Table 3.14: S-Box triple 4,5,6: number of key candidates, total probability and average number of measurements for each collision resistent Δ.

Δ	$\lvert Z\rvert$	$P(collision) = \lvert Z\rvert \cdot 2^{-18}$	$\overline{\#M}$
$\Delta_2 = 000011110010101000$	448	0.00171	1195
$\Delta_5 = 000011111010101000$	896	0.00341	597
$\Delta_8 = 000011111110101000$	448	0.00171	1195
$\Delta_{14} = 000111111010101000$	224	0.00085	3186
$\Delta_{17} = 000111111110101000$	112	0.00043	4778
$\Delta_{20} = 001011110010101000$	224	0.00085	2389
$\Delta_{23} = 001011111010101000$	448	0.00171	1195
$\Delta_{26} = 001011111110101000$	224	0.00085	2389
$\Delta_{29} = 001111110010101000$	224	0.00085	2389
$\Delta_{32} = 001111111010101000$	448	0.00171	1195
$\Delta_{35} = 001111111110101000$	224	0.00085	2389

Table 3.15: S-Box triple 5,6,7: number of key candidates, total probability and average number of measurements for each collision resistent Δ.

Δ	$\lvert Z\rvert$	$P(collision) = \lvert Z\rvert \cdot 2^{-18}$	$\overline{\#M}$
$\Delta_2 = 000111110010101000$	120	0.00046	7680
$\Delta_5 = 000111110110101000$	120	0.00046	7680
$\Delta_7 = 000111111010100100$	384	0.00146	1820
$\Delta_8 = 000111111010101000$	960	0.00366	600
$\Delta_9 = 000111111010101100$	192	0.00073	3641
$\Delta_{10} = 000111111110100100$	96	0.00037	6144
$\Delta_{11} = 000111111110101000$	240	0.00092	2458
$\Delta_{12} = 000111111110101100$	48	0.00018	12288
$\Delta_{19} = 001011111010100100$	256	0.00098	2731
$\Delta_{20} = 001011111010101000$	640	0.00244	900
$\Delta_{21} = 001011111010101100$	128	0.00049	5461
$\Delta_{31} = 001111111010100100$	128	0.00049	5461
$\Delta_{32} = 001111111010101000$	320	0.00122	1800
$\Delta_{33} = 001111111010101100$	64	0.00024	10923

Table 3.16: S-Box triple 6,7,8: number of key candidates, total probability and average number of measurements for each collision resistent Δ.

Δ	$\lvert Z\rvert$	$P(collision) = \lvert Z\rvert \cdot 2^{-18}$	$\overline{\#M}$
$\Delta_1 = 000011110010100100$	576	0.00219	1213
$\Delta_2 = 000011110010101000$	576	0.00219	1277
$\Delta_3 = 000011110010101100$	192	0.00073	4096
$\Delta_4 = 000011110110100100$	384	0.00146	1365
$\Delta_5 = 000011110110101000$	384	0.00146	2456
$\Delta_7 = 000011111010100100$	768	0.00293	682
$\Delta_8 = 000011111010101000$	768	0.00293	1228
$\Delta_{10} = 000011111110100100$	576	0.00219	1213
$\Delta_{11} = 000011111110101000$	576	0.00219	1277
$\Delta_{12} = 000011111110101100$	192	0.00073	4096
$\Delta_{13} = 000111110010100100$	288	0.00110	1820
$\Delta_{14} = 000111110010101000$	288	0.00110	1915
$\Delta_{15} = 000111110010101100$	96	0.00037	6144
$\Delta_{16} = 000111110110100100$	192	0.00073	2731
$\Delta_{17} = 000111110110101000$	192	0.00073	4915
$\Delta_{19} = 000111111010100100$	384	0.00146	1365
$\Delta_{20} = 000111111010101000$	384	0.00146	2457
$\Delta_{22} = 000111111110100100$	288	0.00110	1820
$\Delta_{23} = 000111111110101000$	288	0.00110	1915
$\Delta_{24} = 000111111110101100$	96	0.00037	6144
$\Delta_{28} = 001111110110100100$	96	0.00037	5462
$\Delta_{29} = 001111110110101000$	96	0.00037	9830
$\Delta_{31} = 001111111010100100$	192	0.00073	2730
$\Delta_{32} = 001111111010101000$	192	0.00073	4915

Table 3.17: S-Box triple 7,8,1: number of key candidates, total probability and average number of measurements for each collision resistent Δ.

Δ	$\lvert Z\rvert$	$P(collision) = \lvert Z\rvert \cdot 2^{-18}$	$\overline{\#M}$
$\Delta_1 = 000011110010100100$	240	0.00092	2275
$\Delta_2 = 000011110010101000$	192	0.00073	2730
$\Delta_3 = 000011110010101100$	240	0.00092	2275
$\Delta_4 = 000011110110100100$	120	0.00046	5120
$\Delta_5 = 000011110110101000$	96	0.00036	6144
$\Delta_6 = 000011110110101100$	120	0.00046	5120
$\Delta_7 = 000011111010100100$	360	0.00137	1484
$\Delta_8 = 000011111010101000$	288	0.00110	1843
$\Delta_9 = 000011111010101100$	360	0.00137	1484
$\Delta_{10} = 000011111110100100$	240	0.00092	2194
$\Delta_{11} = 000011111110101000$	192	0.00073	2730
$\Delta_{12} = 000011111110101100$	240	0.00092	2194
$\Delta_{13} = 000111110010100100$	240	0.00092	2274
$\Delta_{14} = 000111110010101000$	192	0.00073	2730
$\Delta_{15} = 000111110010101100$	240	0.00092	2274
$\Delta_{19} = 000111111010100100$	360	0.00137	1684
$\Delta_{20} = 000111111010101000$	288	0.00110	2048
$\Delta_{21} = 000111111010101100$	360	0.00137	1684
$\Delta_{22} = 000111111110100100$	240	0.00092	2275
$\Delta_{23} = 000111111110101000$	192	0.00073	2730
$\Delta_{24} = 000111111110101100$	240	0.00092	2275
$\Delta_{25} = 001011110010100100$	80	0.00031	6827
$\Delta_{26} = 001011110010101000$	64	0.00024	8192
$\Delta_{27} = 001011110010101100$	80	0.00031	6827
$\Delta_{31} = 001011111010100100$	120	0.00046	5052
$\Delta_{32} = 001011111010101000$	96	0.00036	6144
$\Delta_{33} = 001011111010101100$	120	0.00046	5052
$\Delta_{34} = 001011111110100100$	80	0.00031	6827
$\Delta_{35} = 001011111110101000$	64	0.00024	8192
$\Delta_{36} = 001011111110101100$	80	0.00031	6827
$\Delta_{37} = 001111110010100100$	320	0.00122	1707
$\Delta_{38} = 001111110010101000$	256	0.00098	2048
$\Delta_{39} = 001111110010101100$	320	0.00122	1707
$\Delta_{43} = 001111111010100100$	480	0.00185	1263
$\Delta_{44} = 001111111010101000$	384	0.00146	1536
$\Delta_{45} = 001111111010101100$	480	0.00185	1263
$\Delta_{46} = 001111111110100100$	320	0.00122	1707
$\Delta_{47} = 001111111110101000$	256	0.00098	2048
$\Delta_{48} = 001111111110101100$	320	0.00122	1707

Table 3.18: S-Box triple 8,1,2: number of key candidates, total probability and average number of measurements for each collision resistent Δ.

4 Optimization of the Collision Attack

In the last chapter it was shown that it is either possible to minimize the number of plaintext encryptions (and thus the measurement costs) until a first collision occurs or the number of possible key candidates (i.e. the costs of a succeeding brute force attack). The goal of this thesis was to minimize the measurement costs, which means that as few encryptions are required as possible until a collision $f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta)$ occurs. In this chapter additional methods are explained in order to reduce measurement costs and the number of key candidates.

The term collision test was introduced as the comparison of the outputs $f_k(x_{ex})$ and $f_k(x_{ex} \oplus \Delta)$. In practice, this is achieved by correlation of the power traces of the succeeding round. This is explained in detail in Chapter 5. In order to find a collision, an adversary will generate random values of x, i.e. x_{ex}, until he detects a collision $f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta)$. A very simple way to increase the probability of a collision is to keep a history file of generated values x_{ex} and $x_{ex} \oplus \Delta$ in order to ensure that no value of x_{ex} is generated twice.

Thus the probability of a collision slightly increases with each new collision test. If j collision tests have been checked and the corresponding values of x_{ex} and $x_{ex} \oplus \Delta$, for which no collision occured, are not used again, the probability of a collison is

$$P(f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta) | k \in K_i) = \frac{\#possible\ collisions}{2^{14} - 2 \cdot j} \tag{4.1}$$

where the term $\#possible\ collisions$ denotes the number of collisions for a particular differential Δ and a key set K_i (this number of collisions corresponds to an entry in Table 3.10 in Chapter 3). In order to determine the average number of tests until a collision occurs, the probability of each new collision must be added to a continous sum until this sum is greater or equal than one:

$$\sum_{j=0}^{q-1} \frac{\#possible\ collisions}{2^{14} - 2 \cdot j} \geq 1 \tag{4.2}$$

In this summation q denotes the average number of tests until a collision occurs. Since each collision test takes two measurements, the average number of measurements $\overline{\#M}$ is determined as

$$\overline{\#M} = 2 \cdot q \tag{4.3}$$

The improvement of memorizing generated values of x_{ex} and $x_{ex} \oplus \Delta$ in order to reduce the number of collision tests is illustrated in the following example:

> **Example 4.1**: An adversary targets on a particular S-Box triple/Δ combination. He randomly generates values of x until a collision $f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta)$ occurs. The key k is unknown to the adversary. It is assumed that 16 out of 2^{14} values of x_{ex} will cause a collision for this key. If previously generated values of x_{ex} and $x_{ex} \oplus \Delta$ are not

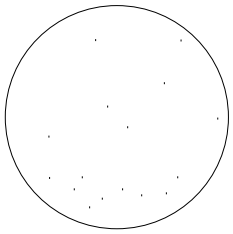


Figure 4.1: 16 out of 2^{14} values of x will cause collisions

> taken into account, the average number of measurements $\overline{\#M}$ until a collision occurs is determined by
>
> $$\overline{\#M} = 2 \cdot \frac{2^{14}}{16} = 2048$$
>
> However, if the adversary makes sure not to check an input x_{ex} twice, the number of measurements is determined by the continous sum:
>
> $$\sum_{j=0}^{q-1} \frac{16}{2^{14} - 2 \cdot j} \geq 1$$
>
> $$\Rightarrow 16 \cdot (\frac{1}{2^{14}} + \frac{1}{2^{14} - 2 \cdot 1} + \ldots + \frac{1}{2^{14} - 2 \cdot 962}) \geq 1$$
>
> Here, $q = 963$ is the least value for which the sum is greater or equal one. Thus, the average number of measurements reduces to
>
> $$\#M = 2 \cdot 963 = 1926$$
>
> which equals a reduction of measurement costs of approximately 6%.

4.1 Multiple Differentials

The attack presented so far targets on a particular S-Box triple and on one fixed Δ. However, it is also possible to use several different Δ's of an S-Box triple simultaneously in order to increase the number of collision tests and thus the total probability of a collision. If n differentials $\Delta_1, \ldots, \Delta_n$ are chosen by an adversary, there exists a set of 2^n possible measurements $f_k(x_{ex})$, $f_k(x_{ex} \oplus \Delta_1)$, $f_k(x_{ex} \oplus \Delta_2)$, $f_k(x_{ex} \oplus \Delta_2 \oplus \Delta_1)$, ..., $f_k(x_{ex} \oplus \Delta_n \oplus \ldots \oplus \Delta_1)$ for each generated value of x_{ex}. Within a set of 2^n measurements, two measurements $f_k(x')$ and $f_k(x'')$ can be checked for a collision, if the x-or difference of x' and x'' equals a known differential $\Delta = x' \oplus x''$ (see Appendix A.3). The following example illustrates the usage of $n = 2$ differentials.

> **Example 4.2**: An adversary chooses a random x (i.e. x_{ex}) and measures $f_k(x_{ex})$ and $f_k(x_{ex} \oplus \Delta_1)$. As shown in Figure 4.2, this will yield collision test A1. Moreover, he chooses an additional differential Δ_2 and measures $f_k(x_{ex} \oplus \Delta_2)$. This will yield collision tests B1. Finally, measuring $f_k(x_{ex} \oplus \Delta_2 \oplus \Delta_1)$ will yield collision tests A2 and B2. (see Figure 4.2).

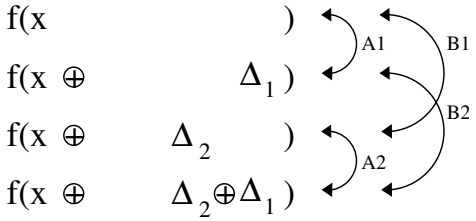


Figure 4.2: Possible collision tests for $n = 2$ differentials Δ_1 and Δ_2.

> Thus a set of $2^n = 2^2 = 4$ measurements will provide four collision tests compared to two collision tests, if only a single Δ was used.

In general, choosing n differentials $\Delta_1, \ldots, \Delta_n$ will yield a set of 2^n measurements $f_k(x_{ex}), f_k(x_{ex} \oplus \Delta_1), f_k(x_{ex} \oplus \Delta_2), f_k(x_{ex} \oplus \Delta_2 \oplus \Delta_1), \ldots, f_k(x_{ex} \oplus \Delta_n \oplus \ldots \oplus \Delta_1)$ for each generated x_{ex} and $n \cdot 2^{n-1}$ collision tests. In Table 4.1 the simple approach of Chapter 3 using a single differential Δ and the advanced approach using n differentials $\Delta_1, \ldots, \Delta_n$ are compared.

The following example stresses the improvement of using $n = 4$ differentials over using a single differential for a fixed number of 128 measurements.

	single Δ	multiple Δ's
$\#x$	m	m
$\#\Delta$	1	n
$\#M$	$2 \cdot m$	$m \cdot 2^n$
$\#collisiontests$	m	$m \cdot n \cdot 2^{n-1}$

Table 4.1: Comparison of the simple and advanced approach of the collision attack

Example 4.3: An adversary uses a single Δ in order to detect a collision $f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta)$. As shown in Table 4.1 $\#M = 128$ measurements will only yield 64 collision tests. As next, it is assumed that the adversary chooses $n = 4$ differentials $\Delta_1, \ldots, \Delta_4$ in order to increase the number of collision tests. For each generated value of x (i.e. x_{ex}), he measures a set of $2^n = 2^4 = 16$ outputs (i.e. power traces of the succeeding round) $f_k(x_{ex}), f_k(x_{ex} \oplus \Delta_1), ..., f_k(x_{ex} \oplus \Delta_4 \oplus \Delta_3 \oplus \Delta_2 \oplus \Delta_1)$ and is able to check $n \cdot 2^{n-1} = 4 \cdot 2^3 = 32$ collision tests. After $m = 8$ generated values of x, he has measured $\#M = 8 \cdot 2^4 = 128$ measurements f_k, and has checked $8 \cdot 4 \cdot 2^3 = 256$ collision tests. The number of collision tests is thus increased by 400% as compared to using a single Δ! Hence, using multiple differentials will efficiently increase the number of collision tests for a fixed number of measurements.

To further illustrate possible collision tests when using multiple $\Delta's$, Figure 4.3 depicts a set of $2^n = 2^3 = 8$ measurements for $n = 3$ differentials Δ_1, Δ_2 and Δ_3. Within a set of 8 measurements, there exist $n \cdot 2^{n-1} = 3 \cdot 2^2 = 12$ collision tests $A1, A2, \ldots, C4$, which are shown as links in Figure 4.3. Depending on the underlying Δ, each collision test is associated with a particular probability. Therefore, collision tests are weighted differently corresponding to the Δ used. The following collision probabilities exist for the set of $2^n = 2^3 = 8$ measurements:

$$\begin{aligned} P(A1) &= P(f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta_1)) \\ P(A2) &= P(f_k(x_{ex} \oplus \Delta_2) = f_k(x_{ex} \oplus \Delta_2 \oplus \Delta_1)) \\ \ldots &= \ldots \\ P(C4) &= P(f_k(x_{ex} \oplus \Delta_2 \oplus \Delta_1) = f_k(x_{ex} \oplus \Delta_3 \oplus \Delta_2 \oplus \Delta_1)) \end{aligned}$$

The collision tests $A1, A2, A3, A4$ and $B1, B2, B3, B4$ and $C1, C2, C3, C4$ are associated with Δ_1, Δ_2 and Δ_3 respectively. Therefore, the corresponding prob-

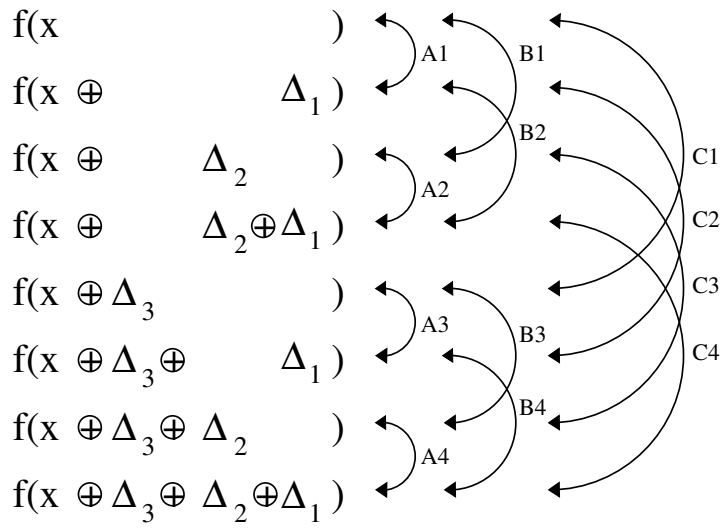


Figure 4.3: Possible collision tests for $n = 3$ differentials Δ_1, Δ_2 and Δ_3

abilities are equal:

$$\begin{aligned} P_1 &= P(A1) = P(A2) = P(A3) = P(A4) = P(f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta_1)) \\ P_2 &= P(B1) = P(B2) = P(B3) = P(B4) = P(f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta_2)) \\ P_3 &= P(C1) = P(C2) = P(C3) = P(C4) = P(f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta_3)) \end{aligned}$$

The overall probability that at least one collision occurs within a set of 8 measurements can be expressed as follows:

$$\begin{aligned} & P((A1 \cup A2 \cup A3 \cup A4) \cup (B1 \cup B2 \cup B3 \cup B4) \cup (C1 \cup C2 \cup C3 \cup C4)) \\ = \; & P(A1) + P(A2) + P(A3) + P(A4) \\ & + P(B1) + P(B2) + P(B3) + P(B4) \\ & + P(C1) + P(C2) + P(C3) + P(C4) \\ & - P(A1 \cap A2 \cap A3 \cap A4 \cap B1 \cap B2 \cap B3 \cap B4 \cap C1 \cap C2 \cap C3 \cap C4) \end{aligned}$$

If collision tests $A1, A2, \ldots, C4$ are *stochastically independent*, i.e. the occurrence of a collision (e.g. A1) does not reflect on the remaining collision tests (A2,...,C4),

the overall probability can also be expressed as:

$$\begin{aligned}
& P((A1 \cup A2 \cup A3 \cup A4) \cup (B1 \cup B2 \cup B3 \cup B4) \cup (C1 \cup C2 \cup C3 \cup C4)) \\
= \; & 1 - P(\overline{(A1 \cup A2 \cup A3 \cup A4) \cup (B1 \cup B2 \cup B3 \cup B4) \cup (C1 \cup C2 \cup C3 \cup C4)}) \\
= \; & 1 - P((\overline{A1} \cap \overline{A2} \cap \overline{A3} \cap \overline{A4}) \cap (\overline{B1} \cap \overline{B2} \cap \overline{B3} \cap \overline{B4}) \cap (\overline{C1} \cap \overline{C2} \cap \overline{C3} \cap \overline{C4})) \\
= \; & 1 - [(1 - P(A1)) \cdot (1 - P(A2)) \cdot (1 - P(A3)) \cdot (1 - P(A4)) \cdot \\
& (1 - P(B1)) \cdot (1 - P(B2)) \cdot (1 - P(B3)) \cdot (1 - P(B4)) \cdot \\
& (1 - P(C1)) \cdot (1 - P(C2)) \cdot (1 - P(C3)) \cdot (1 - P(C4))] \\
\approx \; & P(A1) + P(A2) + \ldots + P(C4)
\end{aligned}$$

In general, if n differentials are being used and there exist no stochastical dependencies among collision tests, the overall probability that at least one collision will occur within a set of 2^n measurements is

$$P(collision) = 1 - (\prod_{i=1}^{n}(1 - P_i))^{2^{n-1}} \approx 2^{n-1} \cdot \sum_{i=1}^{n} P_i \tag{4.4}$$

where P_i denotes the probability of a collision corresponding to a particular Δ_i. The following example illustrates the computation of the overall probability of a collision within a set of $2^n = 2^2 = 4$ measurements for $n = 2$ differentials:

Example 4.4: An adversary tries to cause a collision in S-Boxes 4,5,6 using $n = 2$ differentials Δ_1 and Δ_2. In Table 3.14 (see Chapter 3) the corresponding collision probabilities are stated as $P(f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta_1)) = 0.00122$ and $P(f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta_2)) = 0.00073$. For each set of $2^n = 2^2 = 4$ measurements exist $n \cdot 2^{n-1} = 2 \cdot 2^1 = 4$ collision tests $A1, A2, B1, B2$ with the following probabilities:

$$\begin{aligned}
P(A1) &= P(f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta_1)) = 0.00122 \\
P(A2) &= P(f_k(x_{ex} \oplus \Delta_2) = f_k(x_{ex} \oplus \Delta_2 \oplus \Delta_1)) = P(A1) \\
P(B1) &= P(f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta_2)) = 0.00073 \\
P(B2) &= P(f_k(x_{ex} \oplus \Delta_1) = f_k(x_{ex} \oplus \Delta_2 \oplus \Delta_1)) = P(B1)
\end{aligned}$$

If it is assumed that collision tests $A1$,$A2$,$B1$, $B2$ are stochastically independent, the occurence of a particular collision does not condition any other collision (e.g. if collision A_1 occurs, it is impossible to predict

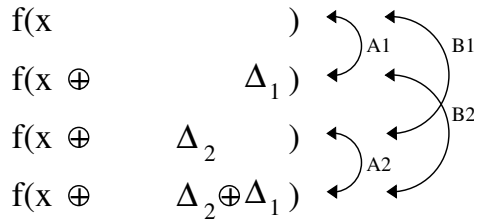


Figure 4.4: Possible collision tests for $n = 2$ differentials Δ_1 and Δ_2

whether any of the remaining collisions A_2, B_1, B_2 will occur):

$$\begin{aligned} P(A1|A2) &= P(A1|B1) = P(A1|B2) = P(A1) \\ P(A2|A1) &= P(A2|B1) = P(A2|B2) = P(A2) \\ P(B1|B2) &= P(B1|A1) = P(B1|A2) = P(B1) \\ P(B2|B1) &= P(B2|A1) = P(B2|A2) = P(B2) \end{aligned}$$

Thus, the probability that all collisions $A1$,$A2$,$B1$, $B2$ occur simultaneously (i.e. the intersection) can be factorized:

$$P(A1 \cap A2 \cap B1 \cap B2) = P(A1) \cdot P(A2) \cdot P(B1) \cdot P(B2) \approx 0$$

The overall probability that at least one collision occurs can then be determined as

$$\begin{aligned} &\Rightarrow \quad P(A1 \cup A2 \cup B1 \cup B2) \\ &\approx \quad P(A1) + P(A2) + P(B1) + P(B2) \\ &= \quad 0.00122 + 0.00122 + 0.00073 + 0.00073 \\ &= \quad 0.0039 \end{aligned}$$

The average number of measurements $\overline{\#M}$ until a collision occurs is found by determining the mean value of measurements $\#M$ over all 256 key sets K_i. Table 4.2 lists an excerpt of the number of collisions of S-Box triple 4,5,6 for the 256 key sets K_i and the input differentials Δ_1 and Δ_2. The average number of measurements $\overline{\#M}$ until a collision

occurs is determined as follows

$$\begin{aligned}\overline{\#M} &= 2^n \cdot \frac{1}{256} \sum_{i=0}^{255} P\left((A1 \cup A2 \cup B1 \cup B2)\,|k \in K_i\right)^{-1} \\ &= 4 \cdot \frac{2^{14}}{256} \left(\frac{1}{(32+32+8+8)} + \frac{1}{(8+8+16+16)} + \ldots + \frac{1}{(32+32+8+8)} \right) \\ &\approx 1092\end{aligned}$$

key set K_i (binary)	Δ_1	Δ_2
xxxx0000xx0000xxxx	32	8
xxxx0000xx0001xxxx	8	16
xxxx0000xx0010xxxx	32	8
xxxx0000xx0011xxxx	8	16
xxxx0000xx0100xxxx	8	16
xxxx0000xx0101xxxx	32	8
xxxx0000xx0110xxxx	8	16
xxxx0000xx0111xxxx	32	8
...	...	...
xxxx1111xx1011xxxx	8	16
xxxx1111xx1100xxxx	8	16
xxxx1111xx1101xxxx	32	8
xxxx1111xx1110xxxx	8	16
xxxx1111xx1111xxxx	32	8

Table 4.2: S-Box triple 4,5,6 and Δ_1,Δ_2: number of collisions for key sets K_i

4.2 Linear and Stochastical Dependencies

Analysis of the Δ-tables in Appendix A.3 reveals that there exist linear combinations of differentials for all eight S-Box triples. If there exist n differentials $\Delta_1, \ldots, \Delta_n$ for a particular S-Box triple, a linear combination is generally expressed as

$$0 = a_1 \cdot \Delta_1 \oplus \ldots \oplus a_n \cdot \Delta_n, a_i \in \{0, 1\} \tag{4.5}$$

A linear combination can then be solved for a particular Δ, e.g.

$$\Delta_1 = a_2 \cdot \Delta_2 \oplus \ldots \oplus a_n \cdot \Delta_n, a_i \in \{0, 1\}$$

Thus it is possible to substitute a particular differential by a linear combination of other differentials. The following example illustrates the linear combination of three $\Delta' s$.

> **Example 4.5**: The linear combination of Δ_3, Δ_{13} and Δ_{15} of S-Box triple 2,3,4 yields Δ_1:
>
> $$\begin{aligned} 0 &= \Delta_1 \oplus \Delta_3 \oplus \Delta_{13} \oplus \Delta_{15} \\ \Leftrightarrow \Delta_1 &= \Delta_3 \oplus \Delta_{13} \oplus \Delta_{15} \end{aligned}$$
>
> $$\begin{array}{lll} & 000011\ 110110\ 101000 & (\Delta_3) \\ \oplus & 000111\ 111010\ 101000 & (\Delta_{13}) \\ \oplus & 000111\ 111110\ 101000 & (\Delta_{15}) \\ \hline = & 000011\ 110010\ 101000 & (\Delta_1) \end{array}$$
>
> Thus Δ_1 can be substituted by the linear combination $\Delta_3 \oplus \Delta_{13} \oplus \Delta_{15}$.

If an adversary targets on a particular S-Box triple and there exist linear combinations among the used differentials, further differentials and thus new collision tests are provided. Thus the overall probability of a collision is increased. This is shown in the following example.

> **Example 4.6**: An adversary tries to cause a collision in an S-Box triple using $n = 3$ differentials Δ_1, Δ_2 and Δ_3. Moreover, there exists the linear combination:
>
> $$\Delta_4 = \Delta_1 \oplus \Delta_2 \oplus \Delta_3$$
>
> As shown in Figure 4.5, this linear combination will provide four additional collision tests $D1, D2, D3$ and $D4$ for every set of $2^n = 2^3 = 8$ measurements. Thus the overall probability of a collision is increased.

The vast improvement that is achieved by linear combinations of differentials is emphasized in the next example.

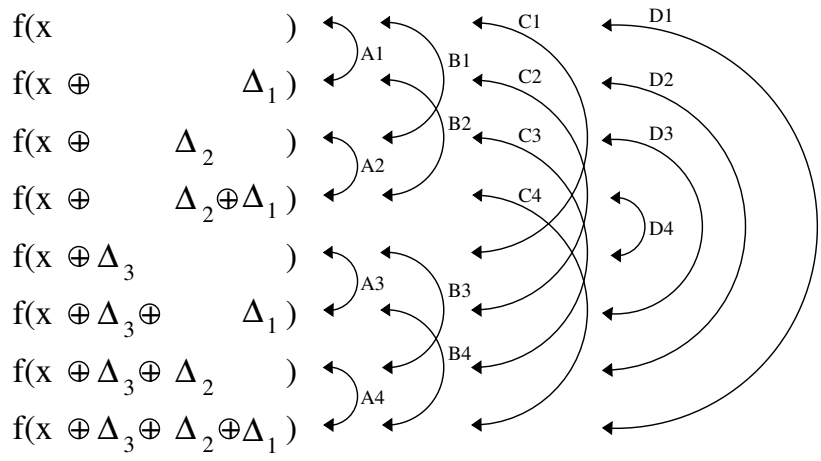


Figure 4.5: Possible collision tests for Δ_1,Δ_2,Δ_3 and $\Delta_4 = \Delta_1 \oplus \Delta_2 \oplus \Delta_3$

Example 4.7: An adversary tries to cause a collision in S-Boxes 2,3,4 using $n = 5$ differentials $\Delta_3, \Delta_{13}, \Delta_{15}, \Delta_{16}$ and Δ_{21} (see Appendix A.4). Analysis of the Δ-table of S-Box triple 2,3,4 reveals that the following linear combinations exist:

$$\begin{aligned}
\Delta_1 &= \Delta_3 \oplus \Delta_{13} \oplus \Delta_{15} \\
\Delta_2 &= \Delta_3 \oplus \Delta_{13} \oplus \Delta_{16} \\
\Delta_4 &= \Delta_3 \oplus \Delta_{15} \oplus \Delta_{16} \\
\Delta_{14} &= \Delta_{13} \oplus \Delta_{15} \oplus \Delta_{16} \\
\Delta_{22} &= \Delta_{15} \oplus \Delta_{16} \oplus \Delta_{21} \\
\Delta_{23} &= \Delta_{13} \oplus \Delta_{15} \oplus \Delta_{21} \\
\Delta_{24} &= \Delta_{13} \oplus \Delta_{16} \oplus \Delta_{21}
\end{aligned}$$

These seven linear combinations will allow the adversary to check $7 \cdot 2^{n-1} = 112$ additional collision tests for each set of $2^n = 32$ measurements. The total number of collision tests for a set of 32 measurements is thus $(n + 7) \cdot 2^{n-1} = 192$. This example also shows that not all differentials, which result from linear combinations, need to be collision resistent[1] (in this case $\Delta_1, \Delta_2, \Delta_4 \Rightarrow$ see Appendix A.3).

In the previous text it has been assumed that collision tests are stochastically independent of each other, i.e. the occurrence of a particular collision within a

[1] i.e. collisions will occur for any key

set of measurements does not condition the remaining collisions. As stated in Chapter 3, for each differential Δ of an S-Box triple exists a collision set Z_Δ of 18 bit input values z. If the elements $z \in Z_\Delta$ of particular differentials Δ are analyzed, it is revealed that stochastical dependencies among collision tests are generally not uncommon. However, stochastical dependent collision tests are not desired, because the they decrease the overall probability of a collision within a set of measurements. The following example further illustrates the meaning of stochastical dependent collision tests.

Example 4.8: An adversary tries to cause a collision in S-Boxes 1,2,3 using $n = 2$ differentials Δ_3, Δ_9. For each set of $2^n = 2^2 = 4$ measure-

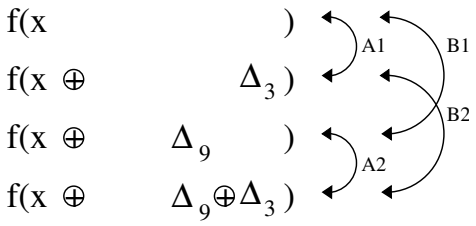


Figure 4.6: Collision tests for $n = 2$ differentials Δ_3 and Δ_9 of S-Box triple 1,2,3

ments exist 4 collision tests: $A1$,$A2$,$B1$ and $B2$. The probabilities of a collision corresponding to Δ_3 and Δ_9 are listed in Table 3.11 (see Chapter 3):

$$\begin{aligned} P(A1) &= P(A2) = P(f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta_3)) \approx 0.00427 \\ P(B1) &= P(B2) = P(f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta_9)) \approx 0.00214 \end{aligned}$$

The probability that at least one collision occurs is

$$\begin{aligned} & P((A1 \cup A2) \cup (B1 \cup B2)) \\ = \; & P(A1) + P(A2) + P(B1) + P(B2) - P((A1 \cap A2) \cap (B1 \cap B2)) \\ = \; & P(A1 \cup A2) + P(B1 \cup B2) - P((A1 \cup A2) \cap (B1 \cup B2)) \\ = \; & P(A1) + P(A2) - P(A1 \cap A2) + P(B1) + P(B2) - P(B1 \cap B2) \\ & -P((A1 \cup A2) \cap (B1 \cup B2)) \end{aligned}$$

Collision set Z_{Δ_3} contains 1120 input values z, which will cause a collision within S-Boxes 1,2,3 (see Appendix A.3). Further analysis of

this particular collision set Z_{Δ_3} reveals that 560 elements z (i.e. % 50) of this set hold the following property:

$$z' = (z \oplus \Delta_9) \in Z_{\Delta_3}$$

This is equivalent to:

$$P(A2|A1) = P(A1|A2) = \frac{P(A1 \cap A2)}{P(A1)} = \frac{P(A2 \cap A1)}{P(A2)} = 0.5$$

This means that if collision $A1$ ($A2$) occurs, the probability is 50% that collision $A2$ ($A1$) occurs as well.

$$\Rightarrow P(A1 \cap A2) = P(A2|A1) \cdot P(A1) \approx 0.5 \cdot 0.00427 = 0.00214$$

Further analysis reveals that no other stochastical dependencies exist for this particular example:

$$\begin{aligned}
P(A1|B1) &= P(A1|B2) = P(A1) \\
P(A2|B1) &= P(A2|B2) = P(A2) \\
P(B1|A1) &= P(B1|A2) = P(B1|B2) = P(B1) \\
P(B2|A1) &= P(B2|A2) = P(B2|B1) = P(B2)
\end{aligned}$$

The total probability of a collision can then be determined as

$$\begin{aligned}
P((A1 \cup A2) \cup (B1 \cup B2)) &= \underbrace{P(A1)}_{\approx 0.00427} + \underbrace{P(A1)}_{\approx 0.00427} - \underbrace{P(A1 \cap A2)}_{\approx 0.00214} \\
&\quad + \underbrace{P(B1)}_{\approx 0.00214} + \underbrace{P(B2)}_{\approx 0.00214} - \underbrace{P(B1 \cap B2)}_{\approx 0} \\
&\quad - \underbrace{P((A1 \cup A2) \cap (B1 \cup B2))}_{\approx 0} \\
&\approx 0.01068
\end{aligned}$$

As a result, the probability that a collision occurs within a set of measurements has decreased due to the stochastical dependency of collision tests $A1$ and $A2$.

$$P((A1 \cup A2) \cup (B1 \cup B2)) = 0.01068 < P(A1) + P(A2) + P(B1) + P(B2) = 0.01282$$

This example shows that stochastically dependent collision tests will decrease the total probability of a collision within a set of 2^n measurements and thus

increase the measurement costs. In general, the possibility that particular collision tests are stochastical dependent increases the more differentials Δ are being used. However, stochastical dependencies and the resulting increase of measurement costs was merely observed in this thesis, but not fully explored. This would be beyond the scope of this work, but the author thinks that stochastical dependencies are closely linked with (yet unknown?) design criteria of the S-Boxes. In order to predict stochastical dependencies, the x-or differences $z' \oplus z''$ among elements $z' \in Z_{\Delta_i}$ and $z'' \in Z_{\Delta_j}$ of a particular S-Box triple must be analyzed[2]. If x-or differences $z' \oplus z''$ equal any existing differential Δ_k and $\Delta_i, \Delta_j, \Delta_k$ are used simultaneously to attack the S-Box triple, collision tests will not be stochatsical independent. As an example, Tables 4.3 to 4.12 list the number of elements $z \in Z_{\Delta_i}$, which satisfy the condition $(z \oplus \Delta_j) \in Z_{\Delta_i}, i \neq j$. These tables indicate that there obviously exist x-or difference patterns among elements $z \in Z_{\Delta}$ and stochastical dependencies are likely to occur. However, the prediction of all stochastical dependencies is a strenuous procedure. Therefore, a computer simulated attack, which exhaustively tries all S-Box triple and Δ combinations, is used to find the optimum combination in order to minimize measurement costs. The results of this exhaustive search are presented in presented in Chapter 6.

[2]the case $i = j$ is valid

Z_Δ	property	$\#z \in Z_\Delta$
Z_{Δ_3}	$z' = (z \oplus \Delta_9) \in Z_{\Delta_3}$	560/1120 (50.00%)
Z_{Δ_3}	$z' = (z \oplus \Delta_{15}) \in Z_{\Delta_3}$	320/1120 (28.57%)
Z_{Δ_3}	$z' = (z \oplus \Delta_{27}) \in Z_{\Delta_3}$	320/1120 (28.57%)
Z_{Δ_3}	$z' = (z \oplus \Delta_{39}) \in Z_{\Delta_3}$	320/1120 (28.57%)
Z_{Δ_6}	$z' = (z \oplus \Delta_9) \in Z_{\Delta_6}$	560/1120 (50.00%)
Z_{Δ_6}	$z' = (z \oplus \Delta_{12}) \in Z_{\Delta_6}$	560/1120 (50.00%)
Z_{Δ_6}	$z' = (z \oplus \Delta_{18}) \in Z_{\Delta_6}$	320/1120 (28.57%)
Z_{Δ_6}	$z' = (z \oplus \Delta_{30}) \in Z_{\Delta_6}$	320/1120 (28.57%)
Z_{Δ_6}	$z' = (z \oplus \Delta_{42}) \in Z_{\Delta_6}$	320/1120 (28.57%)
Z_{Δ_9}	$z' = (z \oplus \Delta_6) \in Z_{\Delta_9}$	560/560 (100.00%)
Z_{Δ_9}	$z' = (z \oplus \Delta_{18}) \in Z_{\Delta_9}$	160/560 (28.57%)
Z_{Δ_9}	$z' = (z \oplus \Delta_{30}) \in Z_{\Delta_9}$	160/560 (28.57%)
Z_{Δ_9}	$z' = (z \oplus \Delta_{42}) \in Z_{\Delta_9}$	160/560 (28.57%)
$Z_{\Delta_{12}}$	$z' = (z \oplus \Delta_6) \in Z_{\Delta_{12}}$	560/560 (100.00%)
$Z_{\Delta_{12}}$	$z' = (z \oplus \Delta_{18}) \in Z_{\Delta_{12}}$	160/560 (28.57%)
$Z_{\Delta_{12}}$	$z' = (z \oplus \Delta_{30}) \in Z_{\Delta_{12}}$	160/560 (28.57%)
$Z_{\Delta_{12}}$	$z' = (z \oplus \Delta_{42}) \in Z_{\Delta_{12}}$	160/560 (28.57%)

Table 4.3: Excerpt of stochastical dependencies of S-Boxes 1,2,3

Z_Δ	property	$\#z \in Z_\Delta$
$Z_{\Delta_{15}}$	$z' = (z \oplus \Delta_9) \in Z_{\Delta_{15}}$	256/512 (50.00%)
$Z_{\Delta_{23}}$	$z' = (z \oplus \Delta_{17}) \in Z_{\Delta_{23}}$	384/768 (50.00%)

Table 4.4: Excerpt of stochastical dependencies of S-Boxes 2,3,4

Z_Δ	property	$\#z \in Z_\Delta$
Z_{Δ_4}	$z' = (z \oplus \Delta_5) \in Z_{\Delta_5}$	128/192 (50.00%)
Z_{Δ_7}	$z' = (z \oplus \Delta_8) \in Z_{\Delta_5}$	128/192 (50.00%)
$Z_{\Delta_{10}}$	$z' = (z \oplus \Delta_{11}) \in Z_{\Delta_5}$	256/384 (25.00%)

Table 4.5: Excerpt of stochastical dependencies of S-Boxes 3,4,5

Z_Δ	property	$\#z \in Z_\Delta$
none	none	none

Table 4.6: Excerpt of stochastical dependencies of S-Boxes 4,5,6

Z_Δ	property	$\#z \in Z_\Delta$
Z_{Δ_5}	$z' = (z \oplus \Delta_8) \in Z_{\Delta_5}$	448/896 (50.00%)
Z_{Δ_5}	$z' = (z \oplus \Delta_{14}) \in Z_{\Delta_5}$	448/896 (50.00%)
Z_{Δ_5}	$z' = (z \oplus \Delta_{17}) \in Z_{\Delta_5}$	224/896 (25.00%)
Z_{Δ_8}	$z' = (z \oplus \Delta_{17}) \in Z_{\Delta_8}$	224/448 (50.00%)
$Z_{\Delta_{14}}$	$z' = (z \oplus \Delta_{17}) \in Z_{\Delta_{14}}$	112/224 (50.00%)
$Z_{\Delta_{20}}$	$z' = (z \oplus \Delta_{23}) \in Z_{\Delta_{20}}$	224/224 (100.00%)
$Z_{\Delta_{23}}$	$z' = (z \oplus \Delta_{20}) \in Z_{\Delta_{23}}$	224/448 (50.00%)
$Z_{\Delta_{23}}$	$z' = (z \oplus \Delta_{26}) \in Z_{\Delta_{23}}$	224/448 (50.00%)
$Z_{\Delta_{29}}$	$z' = (z \oplus \Delta_{32}) \in Z_{\Delta_{29}}$	448/448 (100.00%)
$Z_{\Delta_{32}}$	$z' = (z \oplus \Delta_{29}) \in Z_{\Delta_{32}}$	448/896 (50.00%)
$Z_{\Delta_{32}}$	$z' = (z \oplus \Delta_{35}) \in Z_{\Delta_{32}}$	448/896 (50.00%)

Table 4.7: Excerpt of stochastical dependencies of S-Boxes 5,6,7

Z_Δ	property	$\#z \in Z_\Delta$
Z_{Δ_7}	$z' = (z \oplus \Delta_3) \in Z_{\Delta_7}$	96/384 (25.00%)
Z_{Δ_7}	$z' = (z \oplus \Delta_9) \in Z_{\Delta_7}$	384/384 (100.00%)
Z_{Δ_7}	$z' = (z \oplus \Delta_{10}) \in Z_{\Delta_7}$	192/384 (50.00%)
Z_{Δ_7}	$z' = (z \oplus \Delta_{12}) \in Z_{\Delta_7}$	192/384 (50.00%)
Z_{Δ_8}	$z' = (z \oplus \Delta_3) \in Z_{\Delta_8}$	96/960 (10.00%)
Z_{Δ_8}	$z' = (z \oplus \Delta_5) \in Z_{\Delta_8}$	240/960 (25.00%)
Z_{Δ_8}	$z' = (z \oplus \Delta_9) \in Z_{\Delta_8}$	384/960 (40.00%)
Z_{Δ_8}	$z' = (z \oplus \Delta_{11}) \in Z_{\Delta_8}$	480/960 (50.00%)
Z_{Δ_8}	$z' = (z \oplus \Delta_{12}) \in Z_{\Delta_8}$	192/960 (20.00%)
Z_{Δ_9}	$z' = (z \oplus \Delta_3) \in Z_{\Delta_9}$	48/192 (25.00%)
Z_{Δ_9}	$z' = (z \oplus \Delta_{12}) \in Z_{\Delta_9}$	96/192 (50.00%)
$Z_{\Delta_{10}}$	$z' = (z \oplus \Delta_{12}) \in Z_{\Delta_{10}}$	96/96 (100.00%)
$Z_{\Delta_{11}}$	$z' = (z \oplus \Delta_{12}) \in Z_{\Delta_{11}}$	96/240 (40.00%)
$Z_{\Delta_{19}}$	$z' = (z \oplus \Delta_{21}) \in Z_{\Delta_{19}}$	256/256 (100.00%)
$Z_{\Delta_{20}}$	$z' = (z \oplus \Delta_{21}) \in Z_{\Delta_{20}}$	256/640 (40.00%)
$Z_{\Delta_{31}}$	$z' = (z \oplus \Delta_{33}) \in Z_{\Delta_{31}}$	128/128 (100.00%)
$Z_{\Delta_{32}}$	$z' = (z \oplus \Delta_{33}) \in Z_{\Delta_{32}}$	128/320 (40.00%)

Table 4.8: Excerpt of stochastical deendencies of S-Boxes 6,7,8

Z_Δ	property	$\#z \in Z_\Delta$
Z_{Δ_4}	$z' = (z \oplus \Delta_{28}) \in Z_{\Delta_4}$	192/384 (50.00%)
Z_{Δ_5}	$z' = (z \oplus \Delta_4) \in Z_{\Delta_5}$	256/384 (66.67%)
Z_{Δ_5}	$z' = (z \oplus \Delta_{28}) \in Z_{\Delta_5}$	128/384 (33.33%)
Z_{Δ_5}	$z' = (z \oplus \Delta_{29}) \in Z_{\Delta_5}$	192/384 (50.00%)
Z_{Δ_7}	$z' = (z \oplus \Delta_4) \in Z_{\Delta_7}$	384/768 (50.00%)
Z_{Δ_7}	$z' = (z \oplus \Delta_{10}) \in Z_{\Delta_7}$	384/768 (50.00%)
Z_{Δ_7}	$z' = (z \oplus \Delta_{12}) \in Z_{\Delta_7}$	128/768 (16.67%)
Z_{Δ_7}	$z' = (z \oplus \Delta_{28}) \in Z_{\Delta_7}$	192/768 (25.00%)
Z_{Δ_7}	$z' = (z \oplus \Delta_{31}) \in Z_{\Delta_7}$	384/768 (50.00%)
Z_{Δ_8}	$z' = (z \oplus \Delta_4) \in Z_{\Delta_8}$	256/768 (33.33%)
Z_{Δ_8}	$z' = (z \oplus \Delta_5) \in Z_{\Delta_8}$	384/768 (50.00%)
Z_{Δ_8}	$z' = (z \oplus \Delta_7) \in Z_{\Delta_8}$	512/768 (66.67%)
...	...	...

Table 4.9: Excerpt of stochastical deendencies of S-Boxes 7,8,1

Z_Δ	property	$\#z \in Z_\Delta$
...	...	...
Z_{Δ_8}	$z' = (z \oplus \Delta_{10}) \in Z_{\Delta_8}$	256/768 (33.33%)
Z_{Δ_8}	$z' = (z \oplus \Delta_{11}) \in Z_{\Delta_8}$	384/768 (50.00%)
Z_{Δ_8}	$z' = (z \oplus \Delta_{28}) \in Z_{\Delta_8}$	128/768 (16.67%)
Z_{Δ_8}	$z' = (z \oplus \Delta_{29}) \in Z_{\Delta_8}$	192/768 (25.00%)
Z_{Δ_8}	$z' = (z \oplus \Delta_{31}) \in Z_{\Delta_8}$	256/768 (33.33%)
Z_{Δ_8}	$z' = (z \oplus \Delta_{32}) \in Z_{\Delta_8}$	384/768 (50.00%)
$Z_{\Delta_{10}}$	$z' = (z \oplus \Delta_7) \in Z_{\Delta_{10}}$	384/576 (66.67%)
$Z_{\Delta_{10}}$	$z' = (z \oplus \Delta_{12}) \in Z_{\Delta_{10}}$	192/576 (33.33%)
$Z_{\Delta_{10}}$	$z' = (z \oplus \Delta_{31}) \in Z_{\Delta_{10}}$	192/576 (33.33%)
$Z_{\Delta_{11}}$	$z' = (z \oplus \Delta_7) \in Z_{\Delta_{11}}$	256/576 (44.44%)
$Z_{\Delta_{11}}$	$z' = (z \oplus \Delta_8) \in Z_{\Delta_{11}}$	384/576 (66.67%)
$Z_{\Delta_{11}}$	$z' = (z \oplus \Delta_{10}) \in Z_{\Delta_{11}}$	384/576 (66.67%)
$Z_{\Delta_{11}}$	$z' = (z \oplus \Delta_{31}) \in Z_{\Delta_{11}}$	128/576 (22.22%)
$Z_{\Delta_{11}}$	$z' = (z \oplus \Delta_{32}) \in Z_{\Delta_{11}}$	192/576 (33.33%)
$Z_{\Delta_{13}}$	$z' = (z \oplus \Delta_{15}) \in Z_{\Delta_{13}}$	96/288 (33.33%)
$Z_{\Delta_{13}}$	$z' = (z \oplus \Delta_{19}) \in Z_{\Delta_{13}}$	192/288 (66.67%)
$Z_{\Delta_{14}}$	$z' = (z \oplus \Delta_{13}) \in Z_{\Delta_{14}}$	192/288 (66.67%)
$Z_{\Delta_{14}}$	$z' = (z \oplus \Delta_{19}) \in Z_{\Delta_{14}}$	128/288 (44.44%)
$Z_{\Delta_{14}}$	$z' = (z \oplus \Delta_{20}) \in Z_{\Delta_{14}}$	192/288 (66.67%)
$Z_{\Delta_{17}}$	$z' = (z \oplus \Delta_{16}) \in Z_{\Delta_{17}}$	128/192 (66.67%)
$Z_{\Delta_{19}}$	$z' = (z \oplus \Delta_{16}) \in Z_{\Delta_{19}}$	192/384 (50.00%)
$Z_{\Delta_{19}}$	$z' = (z \oplus \Delta_{22}) \in Z_{\Delta_{19}}$	192/384 (50.00%)
$Z_{\Delta_{19}}$	$z' = (z \oplus \Delta_{24}) \in Z_{\Delta_{19}}$	64/384 (16.67%)
$Z_{\Delta_{20}}$	$z' = (z \oplus \Delta_{16}) \in Z_{\Delta_{20}}$	128/384 (33.33%)
$Z_{\Delta_{20}}$	$z' = (z \oplus \Delta_{17}) \in Z_{\Delta_{20}}$	192/384 (50.00%)
$Z_{\Delta_{20}}$	$z' = (z \oplus \Delta_{19}) \in Z_{\Delta_{20}}$	256/384 (66.67%)
$Z_{\Delta_{20}}$	$z' = (z \oplus \Delta_{22}) \in Z_{\Delta_{20}}$	128/384 (33.33%)
$Z_{\Delta_{20}}$	$z' = (z \oplus \Delta_{23}) \in Z_{\Delta_{20}}$	192/384 (50.00%)
$Z_{\Delta_{22}}$	$z' = (z \oplus \Delta_{19}) \in Z_{\Delta_{22}}$	192/288 (66.67%)
$Z_{\Delta_{22}}$	$z' = (z \oplus \Delta_{24}) \in Z_{\Delta_{22}}$	96/288 (33.33%)
$Z_{\Delta_{23}}$	$z' = (z \oplus \Delta_{19}) \in Z_{\Delta_{23}}$	128/288 (44.44%)
$Z_{\Delta_{23}}$	$z' = (z \oplus \Delta_{20}) \in Z_{\Delta_{23}}$	192/288 (66.67%)
$Z_{\Delta_{23}}$	$z' = (z \oplus \Delta_{22}) \in Z_{\Delta_{23}}$	192/288 (66.67%)
$Z_{\Delta_{29}}$	$z' = (z \oplus \Delta_{28}) \in Z_{\Delta_{29}}$	64/96 (66.67%)
$Z_{\Delta_{31}}$	$z' = (z \oplus \Delta_{28}) \in Z_{\Delta_{31}}$	96/192 (50.00%)
$Z_{\Delta_{32}}$	$z' = (z \oplus \Delta_{28}) \in Z_{\Delta_{32}}$	64/192 (33.33%)
$Z_{\Delta_{32}}$	$z' = (z \oplus \Delta_{29}) \in Z_{\Delta_{32}}$	96/192 (50.00%)
$Z_{\Delta_{32}}$	$z' = (z \oplus \Delta_{31}) \in Z_{\Delta_{32}}$	128/192 (66.67%)

Table 4.10: Excerpt of stochastical dependencies of S-Boxes 7,8,1

Z_Δ	property	$\#z \in Z_\Delta$
Z_{Δ_4}	$z' = (z \oplus \Delta_6) \in Z_{\Delta_4}$	48/120 (40.00%)
Z_{Δ_5}	$z' = (z \oplus \Delta_6) \in Z_{\Delta_5}$	96/96 (100.00%)
Z_{Δ_6}	$z' = (z \oplus \Delta_4) \in Z_{\Delta_6}$	48/120 (40.00%)
Z_{Δ_6}	$z' = (z \oplus \Delta_5) \in Z_{\Delta_6}$	48/120 (40.00%)
Z_{Δ_7}	$z' = (z \oplus \Delta_9) \in Z_{\Delta_7}$	144/360 (40.00%)
Z_{Δ_8}	$z' = (z \oplus \Delta_9) \in Z_{\Delta_8}$	288/288 (100.00%)
Z_{Δ_9}	$z' = (z \oplus \Delta_7) \in Z_{\Delta_9}$	144/360 (40.00%)
Z_{Δ_9}	$z' = (z \oplus \Delta_8) \in Z_{\Delta_9}$	144/360 (40.00%)
$Z_{\Delta_{10}}$	$z' = (z \oplus \Delta_{12}) \in Z_{\Delta_{10}}$	96/240 (40.00%)
$Z_{\Delta_{11}}$	$z' = (z \oplus \Delta_{12}) \in Z_{\Delta_{11}}$	192/192 (100.00%)
$Z_{\Delta_{12}}$	$z' = (z \oplus \Delta_{10}) \in Z_{\Delta_{12}}$	96/240 (40.00%)
$Z_{\Delta_{12}}$	$z' = (z \oplus \Delta_{11}) \in Z_{\Delta_{12}}$	96/240 (40.00%)
$Z_{\Delta_{13}}$	$z' = (z \oplus \Delta_{15}) \in Z_{\Delta_{13}}$	96/240 (40.00%)
$Z_{\Delta_{13}}$	$z' = (z \oplus \Delta_{25}) \in Z_{\Delta_{13}}$	160/240 (66.67%)
$Z_{\Delta_{13}}$	$z' = (z \oplus \Delta_{27}) \in Z_{\Delta_{13}}$	64/240 (26.67%)
$Z_{\Delta_{14}}$	$z' = (z \oplus \Delta_{15}) \in Z_{\Delta_{14}}$	192/192 (100.00%)
$Z_{\Delta_{14}}$	$z' = (z \oplus \Delta_{26}) \in Z_{\Delta_{14}}$	128/192 (66.67%)
$Z_{\Delta_{14}}$	$z' = (z \oplus \Delta_{27}) \in Z_{\Delta_{14}}$	128/192 (66.67%)
$Z_{\Delta_{15}}$	$z' = (z \oplus \Delta_{13}) \in Z_{\Delta_{15}}$	96/240 (40.00%)
$Z_{\Delta_{15}}$	$z' = (z \oplus \Delta_{14}) \in Z_{\Delta_{15}}$	96/240 (40.00%)
$Z_{\Delta_{15}}$	$z' = (z \oplus \Delta_{25}) \in Z_{\Delta_{15}}$	64/240 (26.67%)
$Z_{\Delta_{15}}$	$z' = (z \oplus \Delta_{26}) \in Z_{\Delta_{15}}$	64/240 (26.67%)
$Z_{\Delta_{15}}$	$z' = (z \oplus \Delta_{27}) \in Z_{\Delta_{15}}$	160/240 (66.67%)
$Z_{\Delta_{19}}$	$z' = (z \oplus \Delta_{21}) \in Z_{\Delta_{19}}$	144/360 (40.00%)
$Z_{\Delta_{19}}$	$z' = (z \oplus \Delta_{31}) \in Z_{\Delta_{19}}$	240/360 (66.67%)
$Z_{\Delta_{19}}$	$z' = (z \oplus \Delta_{33}) \in Z_{\Delta_{19}}$	96/360 (26.67%)
$Z_{\Delta_{20}}$	$z' = (z \oplus \Delta_{21}) \in Z_{\Delta_{20}}$	288/288 (100.00%)
$Z_{\Delta_{20}}$	$z' = (z \oplus \Delta_{32}) \in Z_{\Delta_{20}}$	192/288 (66.67%)
$Z_{\Delta_{20}}$	$z' = (z \oplus \Delta_{33}) \in Z_{\Delta_{20}}$	192/288 (66.67%)
$Z_{\Delta_{21}}$	$z' = (z \oplus \Delta_{19}) \in Z_{\Delta_{21}}$	144/360 (40.00%)
$Z_{\Delta_{21}}$	$z' = (z \oplus \Delta_{20}) \in Z_{\Delta_{21}}$	144/360 (40.00%)
$Z_{\Delta_{21}}$	$z' = (z \oplus \Delta_{31}) \in Z_{\Delta_{21}}$	96/360 (26.67%)
$Z_{\Delta_{21}}$	$z' = (z \oplus \Delta_{32}) \in Z_{\Delta_{21}}$	96/360 (26.67%)
$Z_{\Delta_{21}}$	$z' = (z \oplus \Delta_{33}) \in Z_{\Delta_{21}}$	240/360 (66.67%)
$Z_{\Delta_{22}}$	$z' = (z \oplus \Delta_{24}) \in Z_{\Delta_{22}}$	96/240 (40.00%)
$Z_{\Delta_{22}}$	$z' = (z \oplus \Delta_{34}) \in Z_{\Delta_{22}}$	160/240 (66.67%)
$Z_{\Delta_{22}}$	$z' = (z \oplus \Delta_{36}) \in Z_{\Delta_{22}}$	64/240 (26.67%)
...	...	...

Table 4.11: Excerpt of stochastical dependencies of S-Boxes 8,1,2

Z_Δ	property	$\#z \in Z_\Delta$
...	...	...
$Z_{\Delta_{23}}$	$z' = (z \oplus \Delta_{24}) \in Z_{\Delta_{23}}$	192/192 (100.00%)
$Z_{\Delta_{23}}$	$z' = (z \oplus \Delta_{35}) \in Z_{\Delta_{23}}$	128/192 (66.67%)
$Z_{\Delta_{23}}$	$z' = (z \oplus \Delta_{36}) \in Z_{\Delta_{23}}$	128/192 (66.67%)
$Z_{\Delta_{24}}$	$z' = (z \oplus \Delta_{22}) \in Z_{\Delta_{24}}$	96/240 (40.00%)
$Z_{\Delta_{24}}$	$z' = (z \oplus \Delta_{23}) \in Z_{\Delta_{24}}$	96/240 (40.00%)
$Z_{\Delta_{24}}$	$z' = (z \oplus \Delta_{34}) \in Z_{\Delta_{24}}$	64/240 (26.67%)
$Z_{\Delta_{24}}$	$z' = (z \oplus \Delta_{35}) \in Z_{\Delta_{24}}$	64/240 (26.67%)
$Z_{\Delta_{24}}$	$z' = (z \oplus \Delta_{36}) \in Z_{\Delta_{24}}$	160/240 (66.67%)
$Z_{\Delta_{25}}$	$z' = (z \oplus \Delta_{27}) \in Z_{\Delta_{25}}$	32/80 (40.00%)
$Z_{\Delta_{26}}$	$z' = (z \oplus \Delta_{27}) \in Z_{\Delta_{26}}$	64/64 (100.00%)
$Z_{\Delta_{27}}$	$z' = (z \oplus \Delta_{25}) \in Z_{\Delta_{27}}$	32/80 (40.00%)
$Z_{\Delta_{27}}$	$z' = (z \oplus \Delta_{26}) \in Z_{\Delta_{27}}$	32/80 (40.00%)
$Z_{\Delta_{31}}$	$z' = (z \oplus \Delta_{33}) \in Z_{\Delta_{31}}$	48/120 (40.00%)
$Z_{\Delta_{33}}$	$z' = (z \oplus \Delta_{31}) \in Z_{\Delta_{33}}$	48/120 (40.00%)
$Z_{\Delta_{32}}$	$z' = (z \oplus \Delta_{33}) \in Z_{\Delta_{32}}$	96/96 (100.00%)
$Z_{\Delta_{33}}$	$z' = (z \oplus \Delta_{32}) \in Z_{\Delta_{33}}$	48/120 (40.00%)
$Z_{\Delta_{34}}$	$z' = (z \oplus \Delta_{36}) \in Z_{\Delta_{34}}$	32/80 (40.00%)
$Z_{\Delta_{35}}$	$z' = (z \oplus \Delta_{36}) \in Z_{\Delta_{35}}$	64/64 (100.00%)
$Z_{\Delta_{36}}$	$z' = (z \oplus \Delta_{34}) \in Z_{\Delta_{36}}$	32/80 (40.00%)
$Z_{\Delta_{36}}$	$z' = (z \oplus \Delta_{35}) \in Z_{\Delta_{36}}$	32/80 (40.00%)
$Z_{\Delta_{37}}$	$z' = (z \oplus \Delta_{39}) \in Z_{\Delta_{37}}$	128/320 (40.00%)
$Z_{\Delta_{38}}$	$z' = (z \oplus \Delta_{39}) \in Z_{\Delta_{38}}$	256/256 (100.00%)
$Z_{\Delta_{39}}$	$z' = (z \oplus \Delta_{37}) \in Z_{\Delta_{39}}$	128/320 (40.00%)
$Z_{\Delta_{39}}$	$z' = (z \oplus \Delta_{38}) \in Z_{\Delta_{39}}$	128/320 (40.00%)
$Z_{\Delta_{43}}$	$z' = (z \oplus \Delta_{45}) \in Z_{\Delta_{43}}$	192/480 (40.00%)
$Z_{\Delta_{44}}$	$z' = (z \oplus \Delta_{45}) \in Z_{\Delta_{44}}$	384/384 (100.00%)
$Z_{\Delta_{45}}$	$z' = (z \oplus \Delta_{43}) \in Z_{\Delta_{45}}$	192/480 (40.00%)
$Z_{\Delta_{45}}$	$z' = (z \oplus \Delta_{44}) \in Z_{\Delta_{45}}$	192/480 (40.00%)
$Z_{\Delta_{46}}$	$z' = (z \oplus \Delta_{48}) \in Z_{\Delta_{46}}$	128/320 (40.00%)
$Z_{\Delta_{47}}$	$z' = (z \oplus \Delta_{48}) \in Z_{\Delta_{47}}$	256/256 (100.00%)
$Z_{\Delta_{48}}$	$z' = (z \oplus \Delta_{46}) \in Z_{\Delta_{48}}$	128/320 (40.00%)
$Z_{\Delta_{48}}$	$z' = (z \oplus \Delta_{47}) \in Z_{\Delta_{48}}$	128/320 (40.00%)

Table 4.12: Excerpt of stochastical dependencies of S-Boxes 8,1,2

4.3 Key Candidate Reduction

Once a collision $f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta)$ has been detected for a particular differential Δ, there exists a set $K = \{x_{ex} \oplus z | z \in Z_\Delta\}$ of $|Z_\Delta|$ possible 18 bit key candidates. This set must contain the real key. In order to further delimit the number of key candidates and thus decrease the computational costs of a succeeding brute force attack, additional collisions must be found. Each new collision will provide a new key set K_i. Hence, the intersection K_{int} of all sets must contain the real key k:

$$K_{int} = K_1 \cap K_2 \cap \ldots \cap K_j \tag{4.6}$$

Searching for further collisions will always increase the measurement costs. After having found a first collision, an adversary could simply start over again and search for new collisions. If the average number of measurements until a collision is detected is $\overline{\#M}$, the average costs to detect j collisions would then be $j \cdot \overline{\#M}$.

A better approach, which requires only a few additional measurements, can be achieved as follows: instead of causing a new collision in all three S-Boxes, the input of two S-Boxes is fixed and thus the collisions within those S-Boxes are kept. Only the input of the third S-Box is varied until a new collision is found. Due to the bit spreading in the expansion box not all input bits of the third S-Box can be varied in order to find additional collisions. Only bits 2-5 of the left S-Box, bits 2 and 3 of the middle S-Box and bits 0-3 of the right S-Box can be varied without altering the inputs of the other two S-Boxes of the triple.

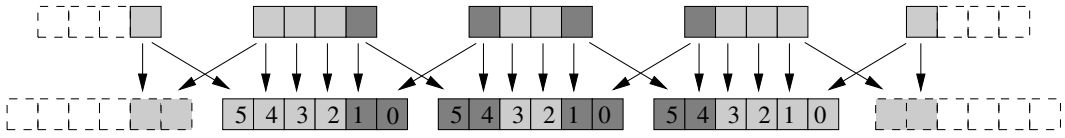


Figure 4.7: Further collisions in single S-Boxes.

Analysis of the collision set Z_Δ provides all existing x-or differences $\epsilon = z' \oplus z''$ between any elements z' and z'' of set Z_Δ. It is possible to cause further collisions in the right, left or middle S-Box, if x-or differences ϵ of the following bit masks exist: $\epsilon = 000000000000000 b_3 b_2 b_1 b_0$, $\epsilon = b_5 b_4 b_3 b_2 000000000000000$ and $\epsilon = 00000000 b_3 b_2 00000000$ with $b_i \in \{0, 1\}$. The maximum number of possibly existing differentials ϵ, which comply with the bits masks stated above, is thus $15 + 15 + 3 = 33$. If such values of ϵ exist within the collision set Z_Δ, further

collisions $f_k(x_{ex} \oplus \epsilon) = f_k(x_{ex} \oplus \epsilon \oplus \Delta)$ might be detected. As an example, Figure 4.8 shows x-or differentials ϵ among eight elements $z \in Z_\Delta$.

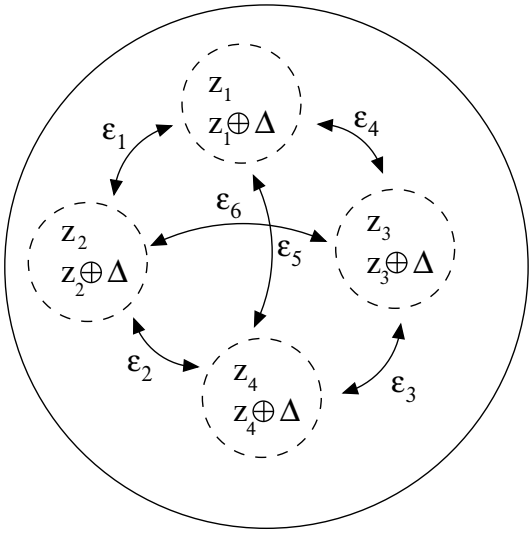


Figure 4.8: Differentials ϵ_i and Δ within a collision set Z_Δ.

The following example illustrates the reduction of key candidates.

Example 4.8: An adversary tries to cause collisions in S-Boxes 1,2,3 using the differential Δ_3. A first collision $f(x_{ex}) = f(x_{ex} \oplus \Delta_3)$ is detected at $x_{ex} = 110110101111111001$. This collision yields $|Z_{\Delta_3}| = 1120$ possible key candidates. Analysis of the particular collision set Z_{Δ_3} reveals that there exist 18 x-or differences $\epsilon_1, \ldots, \epsilon_{18}$, which comply with one of the following bit masks: $\epsilon = 000000000000000 b_3 b_2 b_1 b_0$, $\epsilon = b_5 b_4 b_3 b_2 00000000000000$ or $\epsilon = 00000000 b_3 b_2 00000000$ with $b_i \in \{0, 1\}$. The adversary tries to find further collisions $f_k(x_{ex} \oplus \epsilon_i) = f_k(x_{ex} \oplus \epsilon_i \oplus \Delta_3)$ for all existing differences ϵ_i. Here, as shown by computer simulation, he is able to detect eight additional collisions and delimit[3] the number of key candidates from 1120 down to 16.

[3]By intersection of the key candidate sets provided with each new collision.

5 An Implementation of the Attack

5.1 Computer Simulation

In order to demonstrate the DES collision attack and empirically find the optimum S-Box triple/Δ combination (see Chapter 6.1), a C++ program was developed which simulates the attack on a PC. Initially, it creates the Δ-table of a chosen S-Box triple (see Appendix A.3) and determines the corresponding collision sets Z_Δ. It is then possible to attack this S-Box triple using a particular Δ or a combination of multiple $\Delta's$. The 18 secret key bits of key k can be specified by the user or randomly generated. Moreover, simulated attacks can be iterated with randomized keys in order to obtain mean values, such as the average number of measurements (i.e., encryptions) until a first collision occurs. The program also takes advantage of existing linear combinations of $\Delta's$ in order to decrease measurement costs.

The program was used to double check theoretical results (e.g., predicted collision probabilities), find the optimum Δ combination for each S-Box triple (see Chapter 6) and back up the practical attack against the 8051 microcontroller (see Chapter 5.3).

5.2 Measurement Equipment

If it is assumed that an internal collision is forced in an S-Box triple of round 1 of DES, power analysis of the target hardware running DES can be used to detect such a collision. If a collision occurs, it is assumed that power traces of round 2 will correlate very high due to equal processing. In order to measure the power consumption a small shunt resistance (here $R_s = 10\Omega$) is put in series between the ground pad of the target hardware and ground of the power supply. A digital oscilloscope is used to sample the voltage over the shunt resistance. This

voltage is proportional to the current drawn by the target hardware[1]. In this work the digital oscilloscope HP1662AS was used to measure power traces. The key features of the HP1662AS oscilloscope are:

- 2 channels with 1 GHz sampling rate
- 8 bit Analog Digital Converters (ADCs)
- 250 MHz analog bandwidth in single shot mode
- 8000 samples per channel
- General Purpose Interface Bus (GPIB, IEEE 488-1978) high speed data interface
- Automatic pulse parameters, statistical analysis on measurements

In this work, one channel of the oscilloscope is used for digitizing power traces in single shot mode, the second channel is used for triggering these measurements. The oscilloscope is controlled by a remote PC, which triggers new encryptions and fetches corresponding power traces from the oscilloscope. Programming of the oscillscope and data exchange is realized via the GPIB interface [Pac93]. Here, the Agilent 82357A USB/GPIB interface is used for data transmission between the PC and the oscilloscope. This interface was programmed using the Agilent *Standard Instrument Control Library* (SICL) drivers [Tec01a].

In order to decrease external noise superimposed by the power supply, the power supply of the target hardware can be replaced by a low noise power supply with low voltage ripple. In this work, the Straton 3250.1 DC regulator is used. The key features of this voltage source are:

- Adjustable output voltage 0,005 ... 36 V
- Voltage ripple approximately 1 mV_{eff}
- Voltage stability due to variation of the net voltage max. 1 mV

Uncorrelated noise such as quantization noise of the oscilloscope or intrinsic noise of the semiconductors of the target device can be decreased by averaging

[1] Ohm's law: $V_s = R_s \cdot I$

power traces. As stated in [MDS99], the standard deviation σ' of N averaged power traces is

$$\sigma' = \frac{\sigma}{\sqrt{N}} \tag{5.1}$$

where σ denotes the standard deviation of a single trace due to noise. The averaging of power traces is illustrated in Figures 5.1 to 5.4. These power traces were derived from the 8051 compatible microcontroller OKI MSM80C154S running on a test board. The sampling rate was 1 GHz and the traces were arithmetically averaged using $N = 1, 5, 10$ and $N = 25$ measurements per trace. Comparison of the power traces indicates that averaging will stress certain characteristica, such as distinct peaks, and decrease uncorrelated noise. Since the collision attack proposed in this thesis correlates power traces of different encryptions, these characteristica are essential for the success of the attack. Here, averaging only $N = 5$ measurements will not decrease uncorrelated noise sufficiently. Averaging $N = 10$ measurements results in a power trace, which contains some distinct peaks with reduced noise. Averaging $N = 25$ measurements further reduces noise, but shows only little improvement compared to $N = 10$ measurements. The number of measurements required in order to accurately average power traces strongly depends on the target hardware and the measurement equipment used. In case of the OKI MSM80C154S microcontroller, averaging $N = 10$ measurements is sufficient in order to determine the essential characteristica of a trace.

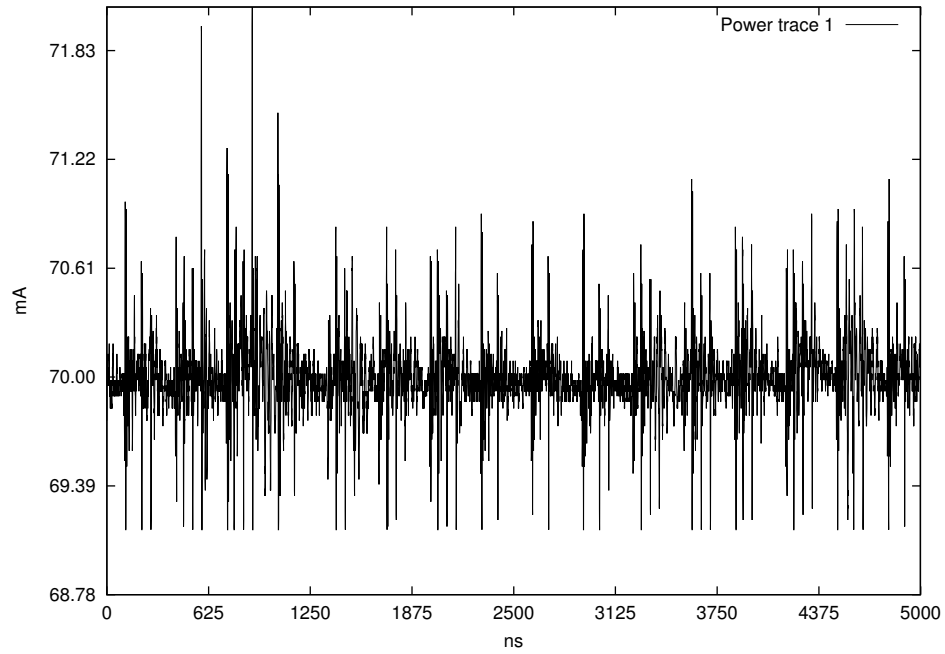


Figure 5.1: Superimposed noise of $N = 1$ measurement

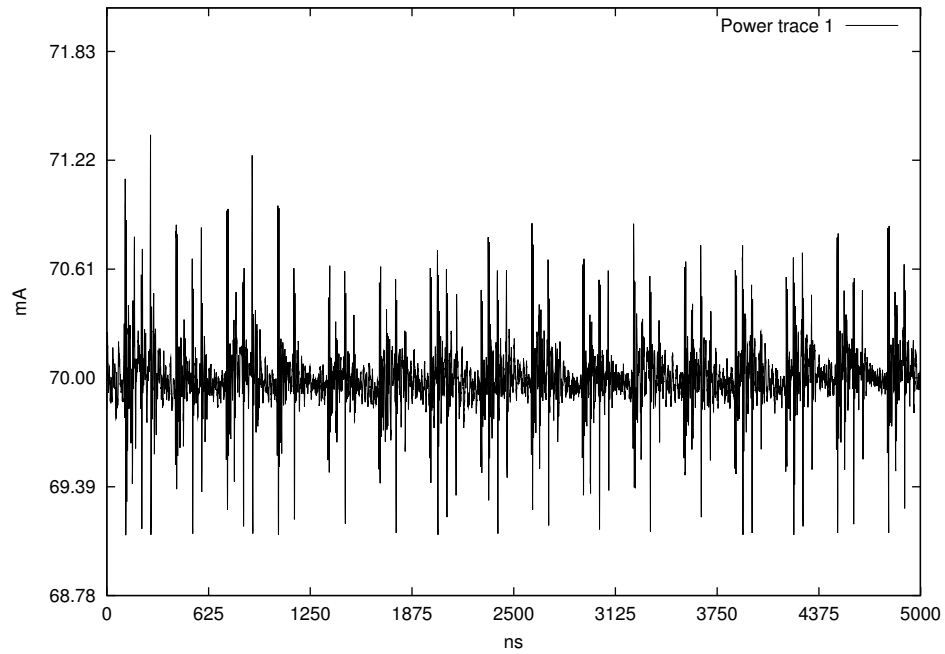


Figure 5.2: Superimposed noise of $N = 5$ averaged measurements

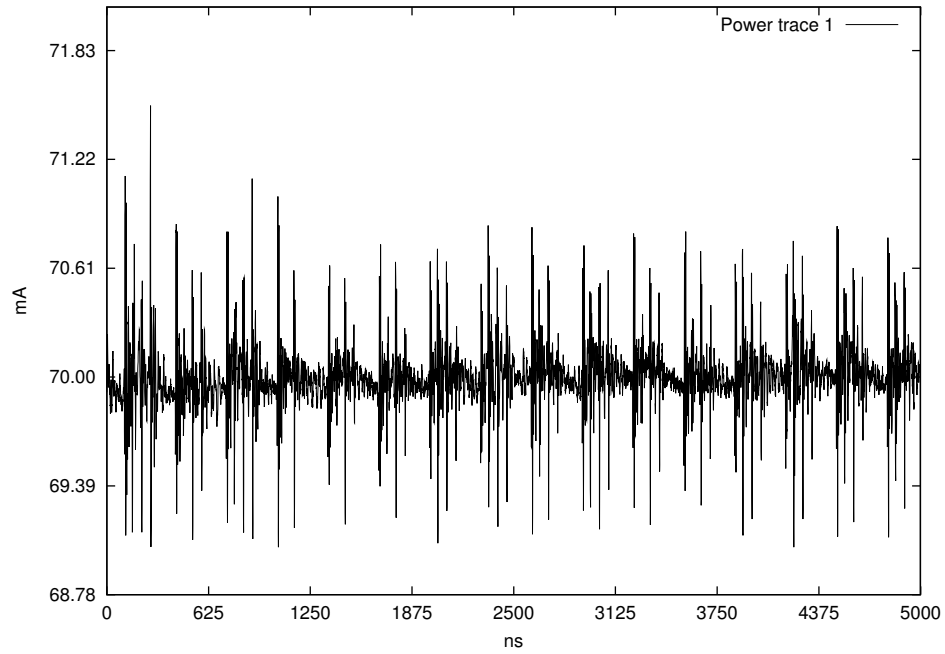


Figure 5.3: Superimposed noise of $N = 10$ averaged measurements

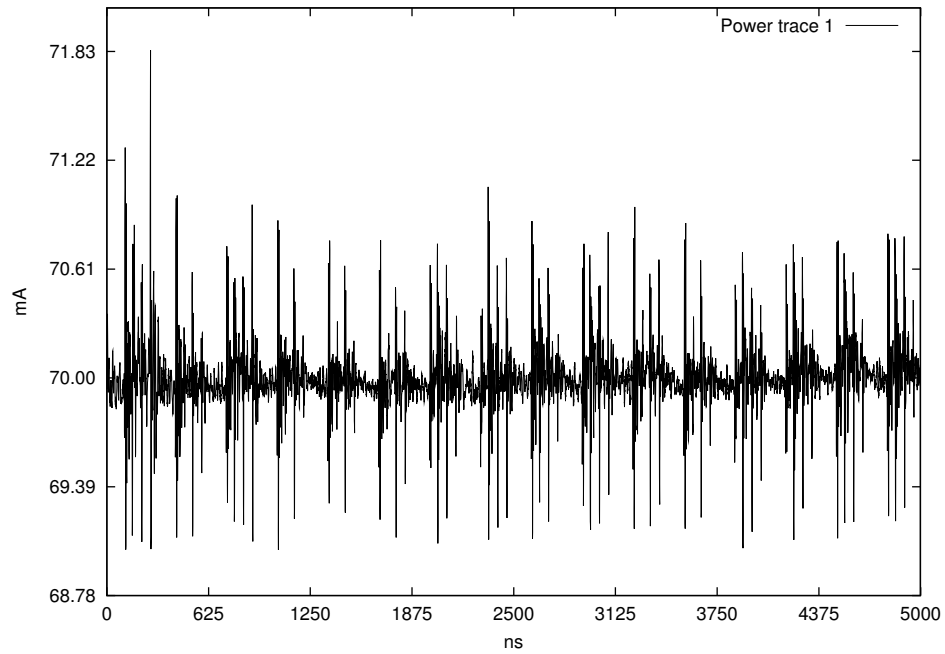


Figure 5.4: Superimposed noise of $N = 25$ averaged measurements

5.3 Compromising DES on an 8051 Microcontroller

In order to demonstrate the practicality of the collision attack, DES running as a software imlementation on an OKI MSM80C154S microcontoller was successfully compromised. The OKI MSM80C154S is an Intel 8051 compatible microcontroller. Derivatives of the 8051 are widely used as a smartcard processors. Therefore, we consider it to be roughly comparable to a typical low-cost smartcard processor without hardware countermeasures against power analysis attacks such as SPA/DPA. A shunt resistance $R_s = 10\Omega$ is put in series between the ground pin of the microcontroller and the external ground node of the low-noise Straton power supply. Channel 1 of the oscilloscope is used to measure the voltage over the shunt resistance (i.e., the current consumption of the microcontroller). Moreover, the operating voltage is increased from $5V$ to approximately $5.6V$ in order to counteract the voltage drop over the shunt resistance.

As part of the thesis, DES was implemented in 8051 assembler. The original 8051 assembler source code of DES was found at [Ser], however it had to be debugged due to bugs in the key scheduling algorithm using an 8051 software simulator. Also, the source code was enhanced with serial communication functions for data transfer with the host PC. Finally, the 8051 assembler code of DES was validated using test vectors supplied in [MvOV97, Sti95].

A program was developed in C++ running on the host PC, which tries to cause collisions in a chosen S-Box triple of round one. The program generates a 14 bit random number x and rearranges the bits within the 64 bit plaintext in order to correctly propagate through the initial permutation (IP) of DES and enter the appropriate S-Box triple. The resulting plaintext is sent via the serial RS232 port at 4600 bps to the microcontroller. The DES algorithm running on the microcontroller has been modified in order to trigger the oscilloscope at the beginning of round two. Once triggered, the oscilloscope waits for a specified time delay and then digitizes a time frame of the power trace of the second round[2]. As stated in Chapter 5.2, it was experimentally determined that averaging of $N = 10$ power traces is sufficient in order to decrease uncorrelated noise and successfully detect collisions. Compared to SPA/DPA related attacks, averaging of only $N = 10$ measurements per power trace is very inexpensive [MS00, CCD00]. This is possible, because only the essential characteristica of a trace need to be

[2]The HP1662AS allows a max. sapling rate of 1GHz.

determined, but no precise data such as hamming weight information [MS00] must be derived from the power traces.

The digital oscilloscope is completely controlled by the program running on the host PC via the GPIB interface. The program requests the digitized power trace from the oscilloscope and stores it in RAM. After encryption of x and transfer of the corresponding power trace, the program encrypts $(x \oplus \Delta)$, requests the corresponding power trace and computes the correlation factor of the two traces. It was shown experimentally that a high correlation factor of approximately 95-100% generally indicates a collision, while a lower correlation factor of 0-80% indicates that different data was processed in round 2 and no collision took place. If the correlation factor lies within 80-100%, the collision detection is extended to several time frames in order to double check whether a true collision occurs in entire round 2 or only in parts of round 2. If a collision was not detected, the program generates a new random x, encrypts x and $(x \oplus \Delta)$, measures the corresponding power traces and repeats the procedure stated above. The measurement setup is shown in Figure 5.5:

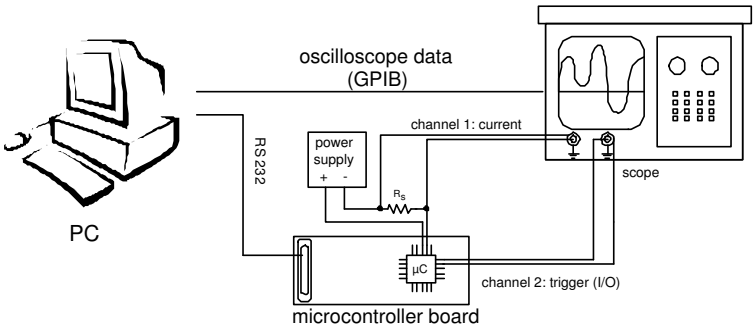


Figure 5.5: Measurement setup for power analysis of a microcontroller.

The clock frequency f_c of the microcontroller is 12 MHz, thus a clock pulse is $t_c = \frac{1}{f_c} \approx 83ns$. The OKI MSM80C154S requires 12 clock pulses per machine cycle, the duration of a one-cycle instruction[3] is $1\mu s$. Power traces are sampled at 1 GHz, thus the sampling period is $10^{-9}s = 1ns$. Lower sampling rates such as 500 MHz or 250 MHz will drastically decrease the accuracy of the collision attack. The HP1662AS oscilloscope features a sample RAM of 8000 points per channel. Thus, the length of a window is $8000ns = 8\mu s$, which corresponds to 8 one-cycle

[3]There exist several instructions, which require two or three machine cycles.

instructions. In Figure 5.6 the duration of function f_k of the 8051 DES implementation is shown in detail. The length of one round equals approximately the

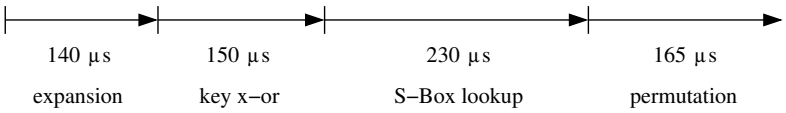


Figure 5.6: Duration of function f_k of the DES software implementation.

length of function f_k, which is $705\mu s$. However, not the full round must be measured and correlated. It is sufficient to focus on the expansion, key x-or addition, S-Box lookup or permutation part of function f_k. Here, the permutation at the end of f_k of round two is analyzed. The reason for this is as follows: if only a few bits change at the ouput of f_k in round one, a false collision during the expansion and key x-or addition might be detected since power traces will correlate very high even though different data is processed. Due to the avalanche effect of the S-Boxes in round 2, collisions can be detected more accurately during the S-Box look up or permutation part of round two. The duration of the permutation part at the end of f_k is approximately $165\mu s$, therefore it is analyzed in order to detect collisions.

The post trigger time delay of the oscilloscope is varied to digitize all succeeding frames of the permutation part. Thus, about $\frac{165\mu s}{8\mu s} \approx 21$ concatenated frames yield the entire power trace of the permutation. However, using up-to-date oscilloscopes with deep memory of up to 32 MB RAM would make this problem obsolete.

Finally, experiments show that it takes approximately 20 minutes to detect a first collision in order to compromise the DES[4] on the microcontroller. However, this duration strongly depends on the measurement equipment used (e.g., oscilloscope probes), amount of oscilloscope memory, underlying key, etc. Using optimized measurement hardware will most probably decrease the duration of the attack to a few minutes.

[4]i.e., exploit details of 18 key bits

5.4 Hardware Countermeasures of a Secure Smartcard

As part of the thesis, a high secure smartcard of the *German Telekom* was also examined. The TeleSec smartcard uses the high secure microcontroller Infineon SLE66CX320P optimized for cryptographic applications (e.g., DES, RSA, digital signatures). Moreover, it features a dedicated cryptographic engine and hardware countermeasures against power analysis attacks [Tec01b]. The measurement setup is shown in Figure 5.5.

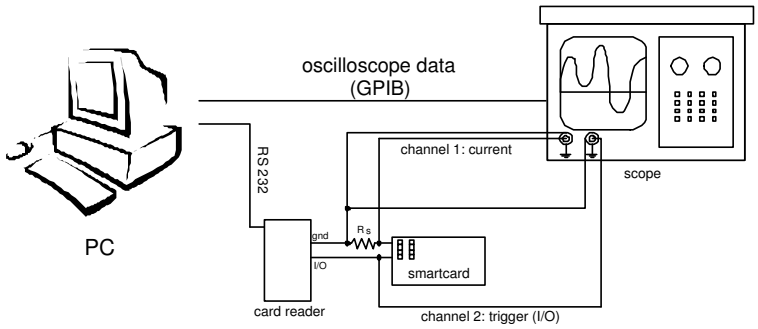


Figure 5.7: Measurement setup for power analysis of a TeleSec smartcard.

Measurements of power traces revealed two significant hardware countermeasures: superimposed noise due to noise generators [LJ01] and *random process interrupts* (RPIs) [CCD00]. Random process interrupts insert dummy cycles at a particular probability. In general, superimposed noise must be decreased by averaging a sufficient number of power traces. However, random process interrupts make it very difficult to arithmetically average power traces, because measured power traces are likely to be uncorrelated due to time shifts. Averaging over uncorrelated traces will generally result in a zero trace.

In order the illustrate the effect of random process interrupts, 10,000 equal DES encryption commands were sent to the TeleSec smartcard. The duration of these DES encryptions[5] is shown in Figure 5.8. The distribution shown in Figure 5.8 clearly resembles a normal (Gaussian) distribution, which implies that random process interrupts occur at constant probability[6] [CCD00]. There exist different methods in digital signal processing, which can be used to exploit this fact.

[5] plus overhead due to the smartcard operating system

[6] central limit theorem

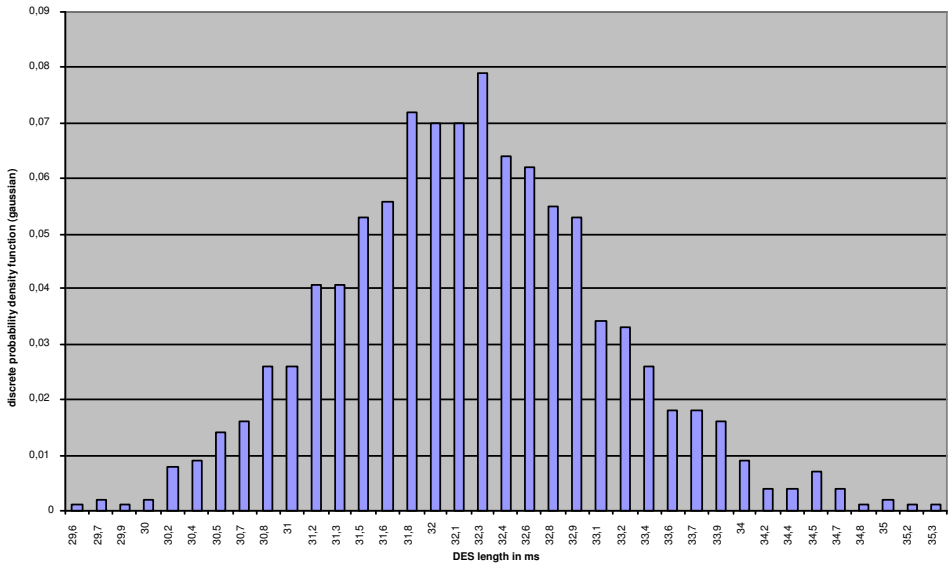


Figure 5.8: Normally distributed durations of equal DES encryption

In [CCD00] a method called *sliding window DPA* is introduced. This method integrates a power trace over a sliding window and makes it possible to successfully perform a DPA on secure hardware. In [Oli00] the principle of a *deconvolution filter* is presented, which can be used to counteract random process interrupts. Moreover, the term *incoherent averaging* is known in digital signal processing as the averaging of desynchronized signals. If these techniques are further studied and adapted to the collision attack in the future, hardware countermeasures of high secure smartcards can most probably be circumvented.

6 Results and Conclusions

In this thesis we developed a new attack against the block cipher DES. The proposed collision attack combines a cryptanalytic approach with power analysis. The main theory of the attack is derived in Chapter 3. Further optimization of the attack is explained in Chapter 4. Chapter 5 briefly describes the practical approach, which was used to demonstrate the attack. As stated in Chapter 5.1, a computer simulated attack was used to exhaustively search for the best S-Box triple / Δ combination in order to minimze measurement costs until a collision occurs. The results of this search are presented in this Chapter. Moreover, further research topics which have to be examined in the future are given.

6.1 Results of our DES collision attack

In Chapter 4 it was shown that it is possible to predict the average number of measurements[1] $\overline{\#M}$ until a collision occurs if all linear and stochastical dependencies of the chosen S-Box triple and $\Delta_1,\ldots,\Delta_n$ combination are known. However, analysis of the corresponding collision sets Z_{Δ_i} in order to derive stochastical dependencies of collision tests becomes complex if several differentials $\Delta_1,\ldots,\Delta_n$ are used. A simpler, but nevertheless reasonable approach exhaustively searches through all possible S-Box triple and differential combinations to find the optimum.

A computer simulated exhaustive search ran on a Pentium IV with 256 MB RAM for ten days. The program used the bubble sort algorithm to iterate through all possible differential combinations for each S-Box triple. Since the number of required measurements until a collision occurs strongly depends on the secret key k (i.e., its key set K_i) each S-Box triple/Δ combination was attacked a 1,000 times with random keys in order to average over all 256 key sets K_i.

The results of the attack are listed in Table 6.1:

[1] equivalent to the number of plaintext encryptions

S-Boxes	$\#\Delta$	$\Delta_1, \Delta_2, \ldots$	$\overline{\#M}$	$\overline{\#K}$
1,2,3	3	$\Delta_3, \Delta_{15}, \Delta_{18}$	227	20
2,3,4	5	$\Delta_3, \Delta_{13}, \Delta_{15}, \Delta_{16}, \Delta_{21}$	140	220
3,4,5	3	$\Delta_3, \Delta_{10}, \Delta_{12}$	190	110
4,5,6	3	$\Delta_2, \Delta_{10}, \Delta_{11}$	690	71
5,6,7	5	$\Delta_2, \Delta_5, \Delta_8, \Delta_{23}, \Delta_{29}$	290	24
6,7,8	5	$\Delta_7, \Delta_{10}, \Delta_{19}, \Delta_{20}, \Delta_{32}$	186	52
7,8,1	5	$\Delta_1, \Delta_2, \Delta_7, \Delta_{17}, \Delta_{19}$	165	19
8,1,2	4	$\Delta_1, \Delta_2, \Delta_8, \Delta_{38}$	208	158

Table 6.1: Results of the exhaustive search for the S-Box triple/Δ set optimum.

The table lists the optimum differential combination for each S-Box triple, which has to be used to minimize the measurement costs $\overline{\#M}$. Furthermore, additional key candidate reduction introduced in Chapter 4 was used. The average number of key candidates $\overline{\#K}$ that was determined for each S-Box triple is also listed in Table 6.1. The minimum number of measurements was found for S-Box triple 2,3,4 and differentials $\Delta_3, \Delta_{13}, \Delta_{15}, \Delta_{16}, \Delta_{21}$. Only an average of $\overline{\#M} = 140$ measurements is required until the first collision occurs. Using key candidate reduction the average number of key candidates $\overline{\#K}$ of this combination can be reduced to 220.

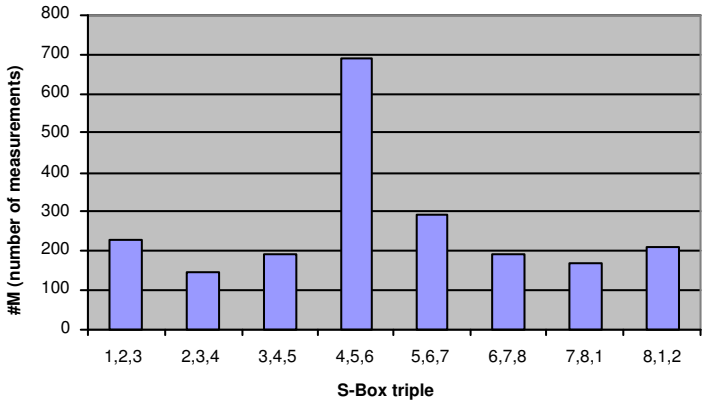


Figure 6.1: Number of measurements for each S-Box triple until a collision occurs for the optimum Δ set.

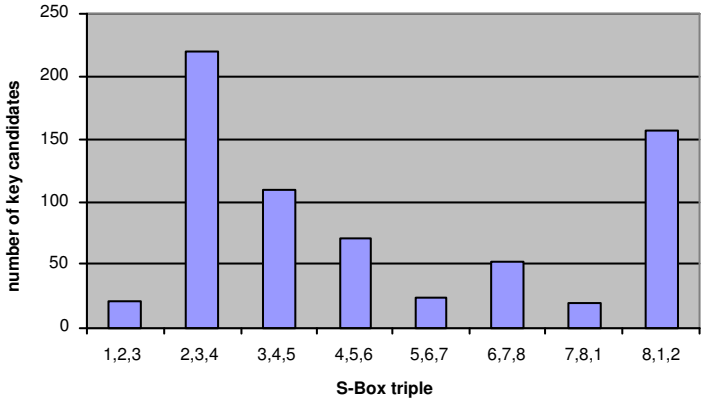


Figure 6.2: Average number of key candidates after key reduction for each S-Box triple and the optimum Δ set.

6.2 Comparison with Wiemers' attack

It is possible to exploit the entire 48 bit round key, if S-Box triples 2,3,4 and 5,6,7 and 7,8,1 are attacked. The average number of encryptions $\overline{\#M}$ in order to cause collisions in these three S-Box triples is $140 + 290 + 165 = 595$. The 48 key bits of the round key can be reduced to an average of $220 + 24 + 19 = 263$ key candidates, which equals approximately $\log_2(263) \approx 8.04$ key bits. As a result, a brute force attack must only search through approximately 16 key bits. Compared to Wiemers' attack the collision attack deduced in this thesis is more expensive with respect to measurement costs. Using Wiemers' attack it is possible to find collisions in three arbitrary S-Boxes with less or equal than $3 \times 9 = 27$ measurements[2], while the minimum number of measurements is 140 if our attack is used. The Wiemers attack provides approximately an average[3] of $(\frac{64}{9})^3 \approx 360$ key candidates for three S-Boxes. Our f_k-collision attack provides between 20 and 220 key candidates, however a direct comparison is not possible, because additional collisions are used during the key reduction method (see Chapter 4.3) to achieve these results. On the other hand an advantage of our attack is that we focus on three S-Boxes which implies that a difference at the output of f_k will have an increased hamming weight. This makes it easier to detect non-collisions in the next round.

[2]averaging of power traces is not considered here
[3]if a S-Box - δ pair is chosen with 9 possible output differentials ϵ_i

As a result, the Wiemers attack is more economic in terms of measurements. It must be investigated in the future whether a combination of both attacks is possible and yields further advantages.

6.3 Future Work

In general, the principle of the collision attack, i.e., the combination of cryptanalysis and side channel analysis, will hold for any cipher, in which internal collisions can be forced and exploited. Therefore, it must be examined, if there exist further cryptographic algorithms which are vulnerable against collision attacks.

Moreover, the stochastical dependencies of collision tests (i.e., the connection between different collision sets Z_Δ of a particular S-Box triple) revealed in Chapter 4 must be examined in more detail. Perhaps unknown details of the S-Boxes will then be discovered.

In Chapter 4.3 a simple method was described in order to find additional collisions and thus delimit the number of key candidates. Certainly, there exist further methods to find additional collisions, which have not been exmined yet:

1. Once a collision has been found for a particular S-Box triple (e.g., 1,2,3), a collision in an overlapping S-Boxe triple (such as 2,3,4 or 8,1,2) will provide information about six additional key bits, but also reduce the number of already known key candidates, because 12 key bits intersect.

2. Once a collision has been found for a particular S-Box triple (e.g., 1,2,3), an adversary could extend the collision attack from three to four S-Boxes. If it is assumed that the attack is extended to the S-Box to the right (e.g., 1,2,3,4), the original 18 bit differential $\Delta = \delta_1|\delta_2|\delta_3$ is replaced by a 24 bit differential $\Delta' = \delta_1|\delta_2|\delta_3'|\delta_4$. Then, x_{ex} denotes a 24 bit input and x the corresponding 18 bit input. The input of the two left S-Boxes (e.g., 1,2) is fixed, i.e., the ten most significant bits of x are not changed. Moreover, the new differential Δ' must fulfil the following conditions:

   a) $\delta_3'[4] = \delta_3[4], \delta_3'[5] = \delta_3[5]$

   b) $\delta_4[5] = \delta_3'[1], \delta_4[4] = \delta_3'[0]$

   c) $\delta_4[0] = \delta_4[1] = 0$

The eight least significant input bits of x can then be varied (i.e., up to $2^8 = 256$ measurements) in order to detect further collisions. Moreover, the determined six bit key candidates corresponding to δ_3 may be exploited to decrease the number of required measurements.

3. Statistical analysis of the collision set Z_Δ can be used to determine all existing differentials ϵ, for which $z' = (z \oplus \epsilon) \in Z_\Delta$. The values of ϵ, which occur with high probability, may then be used to find further collisions $f_k(x_{ex} \oplus \epsilon) = f_k(x_{ex} \oplus \epsilon \oplus \Delta)$.

4. Once a collision $f_k(x_{ex}) = f_k(x_{ex} \oplus \Delta_i)$ has been found, statistical analysis of the collision set Z_{Δ_i}, i.e., its elements $z \in Z_{\Delta_i}$, can be used to determine differentials ϵ, for which $z' = (z \oplus \epsilon) \in Z_{\Delta_j}$. The values of ϵ, which occur with high probability, may then be used to find further collisions $f_k(x_{ex} \oplus \epsilon) = f_k(x_{ex} \oplus \epsilon \oplus \Delta_j)$.

Finally, in Chapter 5 the practical attack was presented. As a side channel the power consumption of the target hardware was analyzed. In the future it must be examined whether analysis of the electromagnetic radiation or both power consumption and radiation provides can be used to perform the collision attack. Using power analysis it was demonstrated that smartcards running DES without hardware countermeasures are vulnerable against the collision attack. However, smartcards with hardware countermeasures such as random process interrupts make it very difficult to average power traces at a particular point of time. Digital signal processing techniques such as incoherent averaging and deconcolution filters [CCD00, Oli00] must be researched in more detail in order to circumvent these countermeasures.

A Appendix

A.1 DES S-Boxes

The eight S-Boxes of DES surjectively map a 6 bit input to a 4 bit output. Bits 5 and 0 of the input contain the row and bits 4 to 1 the column position of the output within the S-Box. The S-Box mappings are uniformly distributed.

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S1	0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
	1	00	15	07	04	14	02	13	01	10	06	12	11	09	05	03	08
	2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
	3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

Table A.1: S-Box 1 of the DES (Data Encryption Standard)

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S2	0	15	01	08	14	06	11	03	04	09	07	02	13	12	00	05	10
	1	03	13	04	07	15	02	08	14	12	00	01	10	06	09	11	05
	2	00	14	07	11	10	04	13	01	05	08	12	06	09	03	02	15
	3	13	08	10	01	03	15	04	02	11	06	07	12	00	05	14	09

Table A.2: S-Box 2 of the DES (Data Encryption Standard)

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S3	0	10	00	09	14	06	03	15	05	01	13	12	07	11	04	02	08
	1	13	07	00	09	03	04	06	10	02	08	05	14	12	11	15	01
	2	13	06	04	09	08	15	03	00	11	01	02	12	05	10	14	07
	3	01	10	13	00	06	09	08	07	04	15	14	03	11	05	02	12

Table A.3: S-Box 3 of the DES (Data Encryption Standard)

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	07	13	14	03	00	06	09	10	01	02	08	05	11	12	04	15
	1	13	08	11	05	06	15	00	03	04	07	02	12	01	10	14	09
S4	2	10	06	09	00	12	11	07	13	15	01	03	14	05	02	08	04
	3	03	15	00	06	10	01	13	08	09	04	05	11	12	07	02	14

Table A.4: S-Box 4 of the DES (Data Encryption Standard)

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	02	12	04	01	07	10	11	06	08	05	03	15	13	00	14	09
	1	14	11	02	12	04	07	13	01	05	00	15	10	03	09	08	06
S5	2	04	02	01	11	10	13	07	08	15	09	12	05	06	03	00	14
	3	11	08	12	07	01	14	02	13	06	15	00	09	10	04	05	03

Table A.5: S-Box 5 of the DES (Data Encryption Standard)

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	12	01	10	15	09	02	06	08	00	13	03	04	14	07	05	11
	1	10	15	04	02	07	12	09	05	06	01	13	14	00	11	03	08
S6	2	09	14	15	05	02	08	12	03	07	00	04	10	01	13	11	06
	3	04	03	02	12	09	05	15	10	11	14	01	07	06	00	08	13

Table A.6: S-Box 6 of the DES (Data Encryption Standard)

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	04	11	02	14	15	00	08	13	03	12	09	07	05	10	06	01
	1	13	00	11	07	04	09	01	10	14	03	05	12	02	15	08	06
S7	2	01	04	11	13	12	03	07	14	10	15	06	08	00	05	09	02
	3	06	11	13	08	01	04	10	07	09	05	00	15	14	02	03	12

Table A.7: S-Box 7 of the DES (Data Encryption Standard)

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	13	02	08	04	06	15	11	01	10	09	03	14	05	00	12	07
	1	01	15	13	08	10	03	07	04	12	05	06	11	00	14	09	02
S8	2	07	11	04	01	09	12	14	02	00	06	10	13	15	03	05	08
	3	02	01	14	07	04	10	08	13	15	12	09	00	03	05	06	11

Table A.8: S-Box 8 of the DES (Data Encryption Standard)

A.2 S-Box δ-tables

The following eight δ-tables list all S-Box inputs z corresponding to existing differentials δ, which fulfil the condition $S_i(z) = S_i(z \oplus \delta)$, with $i = 1, \ldots, 8$. The inputs z are listed in pairs of $(z, z \oplus \delta)$, because both values will fulfil the condition $S_i(z) = S_i(z \oplus \delta)$. Moreover, the position of inputs z within the S-Boxes (e.g. 001000 (4,0)) is given in parentheses for convenience.

δ	#z	(z_1,$z_1 \oplus \delta$), (z_2,$z_2 \oplus \delta$), ...
000011	14	((001000(04,0),001011(05,1)), ((010001(08,1),010010(09,0)), ((010101(10,1),010110(11,0)), ((011000(12,0),011011(13,1)), ((011001(12,1),011010(13,0)), ((100101(02,3),100110(03,2)), ((111001(12,3),111010(13,2))
000101	4	((000010(01,0),000111(03,1)), ((111011(13,3),111110(15,2))
000111	2	((010011(09,1),010100(10,0))
001001	10	((000000(00,0),001001(04,1)), ((000011(01,1),001010(05,0)), ((000100(02,0),001101(06,1)), ((000110(03,0),001111(07,1)), ((100000(00,2),101001(04,3))
001011	2	((100111(03,3),101100(06,2))
001101	6	((010000(08,0),011101(14,1)), ((110001(08,3),111100(14,2)), ((110101(10,3),111000(12,2))
001111	2	((100010(01,2),101101(06,3))
010001	6	((001110(07,0),011111(15,1)), ((100001(00,3),110000(08,2)), ((100011(01,3),110010(09,2))
010011	2	((100100(02,2),110111(11,3))
010111	4	((101000(04,2),111111(15,3)), ((101010(05,2),111101(14,3))
011001	2	((101111(07,3),110110(11,2))
011011	4	((000101(02,1),011110(15,0)), ((001100(06,0),010111(11,1))
011101	4	((000001(00,1),011100(14,0)), ((101110(07,2),110011(09,3))
011111	2	((101011(05,3),110100(10,2))
100010	10	((000010(01,0),100000(00,2)), ((000011(01,1),100001(00,3)), ((001100(06,0),101110(07,2)), ((001111(07,1),101101(06,3)), ((011100(14,0),111110(15,2))
100100	12	((000000(00,0),100100(02,2)), ((000110(03,0),100010(01,2)), ((001000(04,0),101100(06,2)), ((010110(11,0),110010(09,2)), ((010111(11,1),110011(9,3)), ((011000(12,0),111100(14,2))
100101	6	((001101(06,1),101000(04,2)), ((010000(08,0),110101(10,3)), ((011101(14,1),111000(12,2))
.....	..	...

Table A.9: S-Box 1: $S_1(z) = S_1(z \oplus \delta)$

δ	#z	$(z_1, z_1 \oplus \delta)$, $(z_2, z_2 \oplus \delta)$, ...
.....	..	...
100111	10	((000111(03,1),100000(00,2)), ((001011(05,1),101100(06,2)), ((010101(10,1),110010(09,2)), ((011011(13,1),111100(14,2)), ((011100(14,0),111011(13,3))
101000	12	((001110(07,0),100110(03,2)), ((010000(08,0),111000(12,2)), ((010001(08,1),111001(12,3)), ((010010(09,0),111010(13,2)), ((011101(14,1),110101(10,3)), ((011110(15,0),110110(11,2))
101001	4	((010100(10,0),111101(14,3)), ((011000(12,0),110001(08,3))
101010	4	((000101(02,1),101111(07,3)), ((011011(13,1),110001(08,3))
101011	12	((000010(01,0),101001(04,3)), ((000110(03,0),101101(06,3)), ((001010(05,0),100001(00,3)), ((001110(07,0),100101(02,3)), ((010001(8,1),111010(13,2)), ((010010(09,0),111001(12,3))
101100	4	((000100(02,0),101000(04,2)), ((001011(05,1),100111(03,3))
101101	6	((001001(04,1),100100(02,2)), ((001111(07,1),100010(01,2)), ((011001(12,1),110100(10,2))
101110	6	((000111(03,1),101001(04,3)), ((010011(09,1),111101(14,3)), ((011010(13,0),110100(10,2))
101111	2	((001000(04,0),100111(03,3))
110001	4	((011010(13,0),101011(05,3)), ((011110(15,0),101111(07,3))
110010	4	((001101(06,1),111111(15,3)), ((011001(12,1),101011(05,3))
110011	4	((000011(01,1),110000(08,2)), ((000101(02,1),110110(11,2))
110101	2	((010110(11,0),100011(01,3))
110110	2	((010101(10,1),100011(01,3))
110111	2	((000000(00,0),110111(11,3))
111001	6	((010011(09,1),101010(05,2)), ((010111(11,1),101110(07,2)), ((011111(15,1),100110(03,2))
111010	6	((000001(00,1),111011(13,3)), ((001010(05,0),110000(08,2)), ((011111(15,1),100101(02,3))
111011	2	((000100(02,0),111111(15,3))
111110	4	((001001(04,1),110111(11,3)), ((010100(10,0),101010(05,2))
111111	4	((000001(00,1),111110(15,2)), ((001100(06,0),110011(09,3))

Table A.10: S-Box 1: $S_1(z) = S_1(z \oplus \delta)$

δ	#z	$(z_1,z_1 \oplus \delta)$, $(z_2,z_2 \oplus \delta)$, ...
000011	4	((011100(14,0),011111(15,1)), ((110100(10,2),110111(11,3))
000101	2	((110011(9,3),110110(11,2))
000111	4	((101010(5,2),101101(6,3)), ((111000(12,2),111111(15,3))
001001	14	((000000(0,0),001001(4,1)), ((000100(2,0),001101(6,1)), ((000110(3,0),001111(7,1)), ((010001(8,1),011000(12,0)), ((010011(9,1),011010(13,0)), ((010111(11,1),011110(15,0)), ((100111(3,3),101110(7,2))
001011	6	((000101(2,1),001110(7,0)), ((010000(8,0),011011(13,1)), ((110000(8,2),111011(13,3))
001101	6	((000001(0,1),001100(6,0)), ((100001(0,3),101100(6,2)), ((100101(2,3),101000(4,2))
010001	6	((001000(4,0),011001(12,1)), ((100011(1,3),110010(9,2)), ((100100(2,2),110101(10,3))
010011	4	((101001(4,3),111010(13,2)), ((101111(7,3),111100(14,2))
010101	6	((000011(1,1),010110(11,0)), ((000111(3,1),010010(9,0)), ((101011(5,3),111110(15,2))
010111	6	((000010(1,0),010101(10,1)), ((001010(5,0),011101(14,1)), ((100110(3,2),110001(8,3))
011001	2	((100000(0,2),111001(12,3))
011111	4	((001011(5,1),010100(10,0)), ((100010(1,2),111101(14,3))
100010	4	((000011(1,1),100001(0,3)), ((001001(4,1),101011(5,3))
100011	8	((000111(3,1),100100(2,2)), ((001110(7,0),101101(6,3)), ((011010(13,0),111001(12,3)), ((011011(13,1),111000(12,2))
100100	10	((000110(3,0),100010(1,2)), ((001011(5,1),101111(7,3)), ((001110(7,0),101010(5,2)), ((011011(13,1),111111(15,3)), ((011111(15,1),111011(13,3))
100101	6	((000010(1,0),100111(3,3)), ((001100(6,0),101001(4,3)), ((010001(8,1),110100(10,2))
100110	2	((010001(8,1),110111(11,3))
100111	6	((000100(2,0),100011(1,3)), ((010010(9,0),110101(10,3)), ((011100(14,0),111011(13,3))
101000	8	((000001(0,1),101001(4,3)), ((000101(2,1),101101(6,3)), ((010000(8,0),111000(12,2)), ((010100(10,0),111100(14,2))
.....	..	...

Table A.11: S-Box 2: $S_2(z) = S_2(z \oplus \delta)$

δ	#z	(z_1,$z_1 \oplus \delta$), (z_2,$z_2 \oplus \delta$), ...
.....	..	...
101010	4	((010011(9,1),111001(12,3)), ((011001(12,1),110011(9,3))
101011	2	((000000(0,0),101011(5,3))
101100	10	((000010(1,0),101110(7,2)), ((001010(5,0),100110(3,2)), ((011000(12,0),110100(10,2)), ((011100(14,0),110000(8,2)), ((011101(14,1),110001(8,3))
101101	2	((001111(7,1),100010(1,2))
101110	2	((001101(6,1),100011(1,3))
101111	12	((000011(1,1),101100(6,2)), ((000101(2,1),101010(5,2)), ((010000(8,0),111111(15,3)), ((011000(12,0),110111(11,3)), ((011001(12,1),110110(11,2)), ((011111(15,1),110000(8,2))
110010	8	((000111(3,1),110101(10,3)), ((001111(7,1),111101(14,3)), ((010101(10,1),100111(3,3)), ((010111(11,1),100101(2,3))
110011	2	((010011(9,1),100000(0,2))
110110	8	((000100(2,0),110010(9,2)), ((001100(6,0),111010(13,2)), ((010010(9,0),100100(2,2)), ((011110(15,0),101000(4,2))
110111	6	((001001(4,1),111110(15,2)), ((001011(5,1),111100(14,2)), ((010110(11,0),100001(0,3))
111010	4	((010110(11,0),101100(6,2)), ((011010(13,0),100000(0,2))
111011	16	((000001(0,1),111010(13,2)), ((000110(3,0),111101(14,3)), ((001000(4,0),110011(9,3)), ((001010(5,0),110001(8,3)), ((010100(10,0),101111(7,3)), ((010101(10,1),101110(7,2)), ((011101(14,1),100110(3,2)), ((011110(15,0),100101(2,3))
111110	4	((000000(0,0),111110(15,2)), ((001000(4,0),110110(11,2))
111111	4	((001101(6,1),110010(9,2)), ((010111(11,1),101000(4,2))

Table A.12: S-Box 2: $S_2(z) = S_2(z \oplus \delta)$

δ	#z	$(z_1,z_1 \oplus \delta)$, $(z_2,z_2 \oplus \delta)$, ...
000011	8	((000100(2,0),000111(3,1)), ((001001(4,1),001010(5,0)), ((011000(12,0),011011(13,1)), ((111000(12,2),111011(13,3))
000101	6	((001000(4,0),001101(6,1)), ((100000(0,2),100101(2,3)), ((101000(4,2),101101(6,3))
000111	2	((000010(1,0),000101(2,1))
001001	10	((100111(3,3),101110(7,2)), ((110000(8,2),111001(12,3)), ((110100(10,2),111101(14,3)), ((110101(10,3),111100(14,2)), ((110110(11,2),111111(15,3))
001011	2	((100010(1,2),101001(4,3))
001101	8	((010001(8,1),011100(14,0)), ((010011(9,1),011110(15,0)), ((010100(10,0),011001(12,1)), ((100110(3,2),101011(5,3))
001111	4	((000000(0,0),001111(7,1)), ((010000(8,0),011111(15,1))
010001	8	((000110(3,0),010111(11,1)), ((001011(5,1),011010(13,0)), ((001100(6,0),011101(14,1)), ((101111(7,3),111110(15,2))
010011	4	((000001(0,1),010010(9,0)), ((100001(0,3),110010(9,2))
010101	4	((000011(1,1),010110(11,0)), ((100100(2,2),110001(8,3))
011001	4	((100011(1,3),111010(13,2)), ((101010(5,2),110011(9,3))
011011	4	((001110(7,0),010101(10,1)), ((101100(6,2),110111(11,3))
100001	10	((000001(0,1),100000(0,2)), ((000111(3,1),100110(3,2)), ((001000(4,0),101001(4,3)), ((011000(12,0),111001(12,3)), ((011100(14,0),111101(14,3))
100010	12	((000100(2,0),100110(3,2)), ((000101(2,1),100111(3,3)), ((010000(8,0),110010(9,2)), ((010100(10,0),110110(11,2)), ((010111(11,1),110101(10,3)), ((011011(13,1),111001(12,3))
100011	2	((000000(0,0),100011(1,3))
100100	4	((000001(0,1),100101(2,3)), ((001101(6,1),101001(4,3))
100101	6	((000010(1,0),100111(3,3)), ((001001(4,1),101100(6,2)), ((010001(8,1),110100(10,2))
100110	6	((001010(5,0),101100(6,2)), ((001100(6,0),101010(5,2)), ((011001(12,1),111111(15,3))
101000	6	((010110(11,0),111110(15,2)), ((011000(12,0),110000(8,2)), ((011100(14,0),110100(10,2))
101010	2	((001000(4,0),100010(1,2))
.....	..	...

Table A.13: S-Box 3: $S_3(z) = S_3(z \oplus \delta)$

δ	#z	(z_1,$z_1 \oplus \delta$), (z_2,$z_2 \oplus \delta$), ...
.....	..	...
101011	10	((000101(2,1),101110(7,2)), ((010100(10,0),111111(15,3)), ((010111(11,1),111100(14,2)), ((011010(13,0),110001(8,3)), ((011011(13,1),110000(8,2))
101100	10	((000010(1,0),101110(7,2)), ((000011(1,1),101111(7,3)), ((000111(3,1),101011(5,3)), ((001111(7,1),100011(1,3)), ((010001(8,1),111101(14,3))
101101	4	((010101(10,1),111000(12,2)), ((011111(15,1),110010(9,2))
101110	4	((010101(10,1),111011(13,3)), ((011101(14,1),110011(9,3))
101111	8	((000100(2,0),101011(5,3)), ((001011(5,1),100100(2,2)), ((001101(6,1),100010(1,2)), ((011001(12,1),110110(11,2))
110001	2	((010000(8,0),100001(0,3))
110010	2	((010010(9,0),100000(0,2))
110011	4	((000110(3,0),110101(10,3)), ((011110(15,0),101101(6,3))
110101	4	((001110(7,0),111011(13,3)), ((001111(7,1),111010(13,2))
110110	4	((001110(7,0),111000(12,2)), ((011110(15,0),101000(4,2))
110111	4	((010010(9,0),100101(2,3)), ((011101(14,1),101010(5,2))
111001	2	((010110(11,0),101111(7,3))
111010	6	((000000(0,0),111010(13,2)), ((000110(3,0),111100(14,2)), ((001011(5,1),110001(8,3))
111011	2	((010011(9,1),101000(4,2))
111101	4	((000011(1,1),111110(15,2)), ((001010(5,0),110111(11,3))
111110	8	((001001(4,1),110111(11,3)), ((010011(9,1),101101(6,3)), ((011010(13,0),100100(2,2)), ((011111(15,1),100001(0,3))
111111	2	((001100(6,0),110011(9,3))

Table A.14: S-Box 3: $S_3(z) = S_3(z \oplus \delta)$

δ	#z	$(z_1,z_1 \oplus \delta)$, $(z_2,z_2 \oplus \delta)$, ...
000011	8	((000001(0,1),000010(1,0)), ((001001(4,1),001010(5,0)), ((100101(2,3),100110(3,2)), ((101101(6,3),101110(7,2))
000101	4	((001000(4,0),001101(6,1)), ((100010(1,2),100111(3,3))
000111	4	((010010(9,0),010101(10,1)), ((111010(13,2),111101(14,3))
001001	8	((000110(3,0),001111(7,1)), ((010000(8,0),011001(12,1)), ((100000(0,2),101001(4,3)), ((110110(11,2),111111(15,3))
001101	8	((010001(8,1),011100(14,0)), ((010111(11,1),011010(13,0)), ((110011(9,3),111110(15,2)), ((110101(10,3),111000(12,2))
010001	4	((000111(3,1),010110(11,0)), ((101000(4,2),111001(12,3))
010011	8	((000000(0,0),010011(9,1)), ((001100(6,0),011111(15,1)), ((100011(1,3),110000(8,2)), ((101111(7,3),111100(14,2))
010101	8	((001011(5,1),011110(15,0)), ((001110(7,0),011011(13,1)), ((100001(0,3),110100(10,2)), ((100100(2,2),110001(8,3))
010111	4	((000011(1,1),010100(10,0)), ((101100(6,2),111011(13,3))
011001	4	((000100(2,0),011101(14,1)), ((101011(5,3),110010(9,2))
011101	4	((000101(2,1),011000(12,0)), ((101010(5,2),110111(11,3))
100010	8	((010000(8,0),110010(9,2)), ((010001(8,1),110011(9,3)), ((011100(14,0),111110(15,2)), ((011101(14,1),111111(15,3))
100011	4	((010110(11,0),110101(10,3)), ((011010(13,0),111001(12,3))
100111	4	((000110(3,0),100001(0,3)), ((001110(7,0),101001(4,3))
101000	16	((001010(5,0),100010(1,2)), ((001011(5,1),100011(1,3)), ((001100(6,0),100100(2,2)), ((001101(6,1),100101(2,3)), ((010010(9,0),111010(13,2)), ((010011(9,1),111011(13,3)), ((010100(10,0),111100(14,2)), ((010101(10,1),111101(14,3))
101011	8	((001001(4,1),100010(1,2)), ((001101(6,1),100110(3,2)), ((011001(12,1),110010(9,2)), ((011101(14,1),110110(11,2))
101100	8	((000000(0,0),101100(6,2)), ((000001(0,1),101101(6,3)), ((000010(1,0),101110(7,2)), ((000011(1,1),101111(7,3))
101101	4	((001000(4,0),100101(2,3)), ((001010(5,0),100111(3,3))
101110	16	((001000(4,0),100110(3,2)), ((001001(4,1),100111(3,3)), ((001110(7,0),100000(0,2)), ((001111(7,1),100001(0,3)), ((010110(11,0),111000(12,2)), ((010111(11,1),111001(12,3)), ((011110(15,0),110000(8,2)), ((011111(15,1),110001(8,3))
101111	16	((000001(0,1),101110(7,2)), ((000010(1,0),101101(6,3)), ((000101(2,1),101010(5,2)), ((010001(8,1),111110(15,2)), ((010010(9,0),111101(14,3)), ((010101(10,1),111010(13,2)), ((011000(12,0),110111(11,3)), ((011100(14,0),110011(9,3))
110010	16	((000100(2,0),110110(11,2)), ((000101(2,1),110111(11,3)), ((000110(3,0),110100(10,2)), ((000111(3,1),110101(10,3)), ((011000(12,0),101010(5,2)), ((011001(12,1),101011(5,3)), ((011010(13,0),101000(4,2)), ((011011(13,1),101001(4,3))
111011	16	((000000(0,0),111011(13,3)), ((000100(2,0),111111(15,3)), ((001011(5,1),110000(8,2)), ((001111(7,1),110100(10,2)), ((010000(8,0),101011(5,3)), ((010100(10,0),101111(7,3)), ((011011(13,1),100000(0,2)), ((011111(15,1),100100(2,2))
111101	4	((001100(6,0),110001(8,3)), ((011110(15,0),100011(1,3))
111111	8	((000011(1,1),111100(14,2)), ((000111(3,1),111000(12,2)), ((010011(9,1),101100(6,2)), ((010111(11,1),101000(4,2))

Table A.15: S-Box 4: $S_4(z) = S_4(z \oplus \delta)$

δ	#z	$(z_1, z_1 \oplus \delta)$, $(z_2, z_2 \oplus \delta)$, ...
000011	8	((001000(4,0),001011(5,1)), ((010001(8,1),010010(9,0)), ((010101(10,1),010110(11,0)), ((110000(8,2),110011(9,3))
000101	12	((000000(0,0),000101(2,1)), ((000010(1,0),000111(3,1)), ((011011(13,1),011110(15,0)), ((101010(5,2),101111(7,3)), ((110010(9,2),110111(11,3)), ((111010(13,2),111111(15,3))
000111	2	((100001(0,3),100110(3,2))
001001	8	((000110(3,0),001111(7,1)), ((010011(9,1),011010(13,0)), ((110001(8,3),111000(12,2)), ((110101(10,3),111100(14,2))
001011	4	((100111(3,3),101100(6,2)), ((110110(11,2),111101(14,3))
001101	10	((000100(2,0),001001(4,1)), ((010000(8,0),011101(14,1)), ((010100(10,0),011001(12,1)), ((100011(1,3),101110(7,2)), ((100100(2,2),101001(4,3))
001111	4	((000011(1,1),001100(6,0)), ((100010(1,2),101101(6,3))
010001	6	((001110(7,0),011111(15,1)), ((100101(2,3),110100(10,2)), ((101000(4,2),111001(12,3))
010101	4	((001101(6,1),011000(12,0)), ((101011(5,3),111110(15,2))
011011	2	((100000(0,2),111011(13,3))
011101	4	((000001(0,1),011100(14,0)), ((001010(5,0),010111(11,1))
100010	14	((000000(0,0),100010(1,2)), ((000011(1,1),100001(0,3)), ((000110(3,0),100100(2,2)), ((000111(3,1),100101(2,3)), ((001010(5,0),101000(4,2)), ((001101(6,1),101111(7,3)), ((011100(14,0),111110(15,2))
100011	2	((011001(12,1),111010(13,2))
100100	6	((000100(2,0),100000(0,2)), ((001000(4,0),101100(6,2)), ((010010(9,0),110110(11,2))
100101	6	((000011(1,1),100110(3,2)), ((010101(10,1),110000(8,2)), ((010110(11,0),110011(9,3))
100110	12	((001111(7,1),101001(4,3)), ((010011(9,1),110101(10,3)), ((010101(10,1),110011(9,3)), ((010110(11,0),110000(8,2)), ((011001(12,1),111111(15,3)), ((011010(13,0),111100(14,2))
100111	12	((000010(1,0),100101(2,3)), ((000101(2,1),100010(1,2)), ((001011(5,1),101100(6,2)), ((001101(6,1),101010(5,2)), ((010001(8,1),110110(11,2)), ((011111(15,1),111000(12,2))
101000	2	((000101(2,1),101101(6,3))
101001	6	((001001(4,1),100000(0,2)), ((011011(13,1),110010(9,2)), ((011110(15,0),110111(11,3))
101010	4	((000001(0,1),101011(5,3)), ((001100(6,0),100110(3,2))
.....	..	...

Table A.16: S-Box 5: $S_5(z) = S_5(z \oplus \delta)$

δ	#z	$(z_1, z_1 \oplus \delta)$, $(z_2, z_2 \oplus \delta)$, ...
.....	..	...
101011	4	((001111(7,1),100100(2,2)), ((010100(10,0),111111(15,3))
101100	8	((001011(5,1),100111(3,3)), ((010001(8,1),111101(14,3)), ((011011(13,1),110111(11,3)), ((011110(15,0),110010(9,2))
101101	4	((000000(0,0),101101(6,3)), ((001100(6,0),100001(0,3))
101110	6	((010100(10,0),111010(13,2)), ((010111(11,1),111001(12,3)), ((011111(15,1),110001(8,3))
101111	10	((000110(3,0),101001(4,3)), ((001000(4,0),100111(3,3)), ((010010(9,0),111101(14,3)), ((010011(9,1),111100(14,2)), ((011010(13,0),110101(10,3))
110010	4	((001001(4,1),111011(13,3)), ((011000(12,0),101010(5,2))
110011	8	((000111(3,1),110100(10,2)), ((001010(5,0),111001(12,3)), ((010000(8,0),100011(1,3)), ((011101(14,1),101110(7,2))
110110	4	((000010(1,0),110100(10,2)), ((001110(7,0),111000(12,2))
110111	4	((011000(12,0),101111(7,3)), ((011100(14,0),101011(5,3))
111110	4	((010000(8,0),101110(7,2)), ((011101(14,1),100011(1,3))
111111	8	((000001(0,1),111110(15,2)), ((000100(2,0),111011(13,3)), ((001110(7,0),110001(8,3)), ((010111(11,1),101000(4,2))

Table A.17: S-Box 5: $S_5(z) = S_5(z \oplus \delta)$

δ	#z	(z_1,$z_1 \oplus \delta$), (z_2,$z_2 \oplus \delta$), ...
000101	10	((000001(0,1),000100(2,0)), ((000011(1,1),000110(3,0)), ((001000(4,0),001101(6,1)), ((011011(13,1),011110(15,0)), ((111010(13,2),111111(15,3))
000111	6	((010010(9,0),010101(10,1)), ((110000(8,2),110111(11,3)), ((111001(12,3),111110(15,2))
001001	10	((010000(8,0),011001(12,1)), ((010100(10,0),011101(14,1)), ((100000(0,2),101001(4,3)), ((100100(2,2),101101(6,3)), ((110010(9,2),111011(13,3))
001011	4	((000000(0,0),001011(5,1)), ((100111(3,3),101100(6,2))
001101	12	((000111(3,1),001010(5,0)), ((100011(1,3),101110(7,2)), ((100101(2,3),101000(4,2)), ((100110(3,2),101011(5,3)), ((110001(8,3),111100(14,2)), ((110101(10,3),111000(12,2))
001111	2	((010111(11,1),011000(12,0))
010001	6	((000010(1,0),010011(9,1)), ((001110(7,0),011111(15,1)), ((100010(1,2),110011(9,3))
010011	6	((000101(2,1),010110(11,0)), ((001001(4,1),011010(13,0)), ((001111(7,1),011100(14,0))
010101	2	((100001(0,3),110100(10,2))
010111	2	((101010(5,2),111101(14,3))
011001	2	((101111(7,3),110110(11,2))
011101	2	((001100(6,0),010001(8,1))
100001	2	((001000(4,0),101001(4,3))
100010	16	((000110(3,0),100100(2,2)), ((000111(3,1),100101(2,3)), ((001010(5,0),101000(4,2)), ((010000(8,0),110010(9,2)), ((010110(11,0),110100(10,2)), ((011001(12,1),111011(13,3)), ((011110(15,0),111100(14,2)), ((011111(15,1),111101(14,3))
100100	10	((000101(2,1),100001(0,3)), ((001101(6,1),101001(4,3)), ((001110(7,0),101010(5,2)), ((001111(7,1),101011(5,3)), ((010111(11,1),110011(9,3))
100110	2	((010011(9,1),110101(10,3))
100111	8	((000000(0,0),100111(3,3)), ((000011(1,1),100100(2,2)), ((001011(5,1),101100(6,2)), ((011011(13,1),111100(14,2))
101000	6	((001000(4,0),100000(0,2)), ((010001(8,1),111001(12,3)), ((010010(9,0),111010(13,2))
.....	..	...

Table A.18: S-Box 6: $S_6(z) = S_6(z \oplus \delta)$

δ	#z	$(z_1, z_1 \oplus \delta)$, $(z_2, z_2 \oplus \delta)$, ...
.....	..	...
101001	2	((001111(7,1),100110(3,2))
101010	6	((010101(10,1),111111(15,3)), ((011010(13,0),110000(8,2)), ((011011(13,1),110001(8,3))
101011	12	((000100(2,0),101111(7,3)), ((000110(3,0),101101(6,3)), ((010000(8,0),111011(13,3)), ((010011(9,1),111000(12,2)), ((011000(12,0),110011(9,3)), ((011001(12,1),110010(9,2))
101100	4	((000000(0,0),101100(6,2)), ((001011(5,1),100111(3,3))
101101	6	((001101(6,1),100000(0,2)), ((010010(9,0),111111(15,3)), ((011010(13,0),110111(11,3))
101110	4	((000001(0,1),101111(7,3)), ((000011(1,1),101101(6,3))
101111	10	((000111(3,1),101000(4,2)), ((001010(5,0),100101(2,3)), ((010001(8,1),111110(15,2)), ((010101(10,1),111010(13,2)), ((011110(15,0),110001(8,3))
110001	2	((000101(2,1),110100(10,2))
110010	4	((000100(2,0),110110(11,2)), ((001100(6,0),111110(15,2))
110011	4	((001110(7,0),111101(14,3)), ((011101(14,1),101110(7,2))
110101	6	((001100(6,0),111001(12,3)), ((010111(11,1),100010(1,2)), ((011111(15,1),101010(5,2))
110111	10	((000001(0,1),110110(11,2)), ((000010(1,0),110101(10,3)), ((010100(10,0),100011(1,3)), ((010110(11,0),100001(0,3)), ((011100(14,0),101011(5,3))
111001	2	((001001(4,1),110000(8,2))
111010	8	((000010(1,0),111000(12,2)), ((010100(10,0),101110(7,2)), ((011000(12,0),100010(1,2)), ((011100(14,0),100110(3,2))
111110	4	((001001(4,1),110111(11,3)), ((011101(14,1),100011(1,3))

Table A.19: S-Box 6: $S_6(z) = S_6(z \oplus \delta)$

δ	#z	$(z_1,z_1 \oplus \delta)$, $(z_2,z_2 \oplus \delta)$, ...
000011	8	((010000(8,0),010011(9,1)), ((011100(14,0),011111(15,1)), ((100101(2,3),100110(3,2)), ((101100(6,2),101111(7,3))
000101	6	((010010(9,0),010111(11,1)), ((110010(9,2),110111(11,3)), ((111011(13,3),111110(15,2))
000111	4	((000010(1,0),000101(2,1)), ((100011(1,3),100100(2,2))
001001	10	((000000(0,0),001001(4,1)), ((000011(1,1),001010(5,0)), ((100000(0,2),101001(4,3)), ((100010(1,2),101011(5,3)), ((110011(9,3),111010(13,2))
001101	6	((010101(10,1),011000(12,0)), ((110001(8,3),111100(14,2)), ((110101(10,3),111000(12,2))
001111	2	((000001(0,1),001110(7,0))
010001	6	((000111(3,1),010110(11,0)), ((001100(6,0),011101(14,1)), ((100111(3,3),110110(11,2))
010011	4	((001000(4,0),011011(13,1)), ((001101(6,1),011110(15,0))
010101	4	((001111(7,1),011010(13,0)), ((100001(0,3),110100(10,2))
010111	8	((000110(3,0),010001(8,1)), ((101000(4,2),111111(15,3)), ((101010(5,2),111101(14,3)), ((101110(7,2),111001(12,3))
011101	4	((000100(2,0),011001(12,1)), ((101101(6,3),110000(8,2))
011111	2	((001011(5,1),010100(10,0))
100001	4	((000010(1,0),100011(1,3)), ((000101(2,1),100100(2,2))
100010	10	((000000(0,0),100010(1,2)), ((001001(4,1),101011(5,3)), ((001111(7,1),101101(6,3)), ((011000(12,0),111010(13,2)), ((011001(12,1),111011(13,3))
100100	4	((000001(0,1),100101(2,3)), ((001101(6,1),101001(4,3))
100101	2	((010100(10,0),110001(8,3))
.....	..	...

Table A.20: S-Box 7: $S_7(z) = S_7(z \oplus \delta)$

δ	#z	$(z_1, z_1 \oplus \delta)$, $(z_2, z_2 \oplus \delta)$, ...
.....	..	...
100110	6	((000010(1,0),100100(2,2)), ((000101(2,1),100011(1,3)), ((010101(10,1),110011(9,3))
100111	4	((000001(0,1),100110(3,2)), ((011001(12,1),111110(15,2))
101000	14	((000110(3,0),101110(7,2)), ((000111(3,1),101111(7,3)), ((001110(7,0),100110(3,2)), ((010001(8,1),111001(12,3)), ((010100(10,0),111100(14,2)), ((010111(11,1),111111(15,3)), ((011100(14,0),110100(10,2))
101001	2	((011011(13,1),110010(9,2))
101010	2	((011010(13,0),110000(8,2))
101011	16	((000000(0,0),101011(5,3)), ((000111(3,1),101100(6,2)), ((001001(4,1),100010(1,2)), ((001100(6,0),100111(3,3)), ((001110(7,0),100101(2,3)), ((011000(12,0),110011(9,3)), ((011101(14,1),110110(11,2)), ((011111(15,1),110100(10,2))
101100	2	((011011(13,1),110111(11,3))
101101	6	((001101(6,1),100000(0,2)), ((010000(8,0),111101(14,3)), ((010010(9,0),111111(15,3))
101110	2	((010011(9,1),111101(14,3))
101111	2	((010101(10,1),111010(13,2))
110010	2	((001010(5,0),111000(12,2))
110110	2	((000011(1,1),110101(10,3))
110111	6	((001011(5,1),111100(14,2)), ((011010(13,0),101101(6,3)), ((011110(15,0),101001(4,3))
111001	4	((010011(9,1),101010(5,2)), ((010110(11,0),101111(7,3))
111010	16	((000100(2,0),111110(15,2)), ((001000(4,0),110010(9,2)), ((001011(5,1),110001(8,3)), ((001100(6,0),110110(11,2)), ((010000(8,0),101010(5,2)), ((010010(9,0),101000(4,2)), ((010110(11,0),101100(6,2)), ((011101(14,1),100111(3,3))
111011	2	((000011(1,1),111000(12,2))
111101	2	((011100(14,0),100001(0,3))
111110	4	((011110(15,0),100000(0,2)), ((011111(15,1),100001(0,3))
111111	14	((000100(2,0),111011(13,3)), ((000110(3,0),111001(12,3)), ((001000(4,0),110111(11,3)), ((001010(5,0),110101(10,3)), ((001111(7,1),110000(8,2)), ((010001(8,1),101110(7,2)), ((010111(11,1),101000(4,2))

Table A.21: S-Box 7: $S_7(z) = S_7(z \oplus \delta)$

δ	#z	$(z_1,z_1 \oplus \delta)$, $(z_2,z_2 \oplus \delta)$, ...
000011	6	((000100(2,0),000111(3,1)), ((011001(12,1),011010(13,0)), ((111001(12,3),111010(13,2))
000101	4	((000000(0,0),000101(2,1)), ((100011(1,3),100110(3,2))
000111	6	((100000(0,2),100111(3,3)), ((110000(8,2),110111(11,3)), ((111011(13,3),111100(14,2))
001001	8	((000011(1,1),001010(5,0)), ((000110(3,0),001111(7,1)), ((100101(2,3),101100(6,2)), ((110001(8,3),111000(12,2))
001011	2	((010011(9,1),011000(12,0))
001101	6	((010001(8,1),011100(14,0)), ((010110(11,0),011011(13,1)), ((100100(2,2),101001(4,3))
001111	8	((000001(0,1),001110(7,0)), ((010010(9,0),011101(14,1)), ((100001(0,3),101110(7,2)), ((110010(9,2),111101(14,3))
010011	4	((001101(6,1),011110(15,0)), ((101101(6,3),111110(15,2))
011001	6	((001001(4,1),010000(8,0)), ((101010(5,2),110011(9,3)), ((101111(7,3),110110(11,2))
011011	2	((001100(6,0),010111(11,1))
011101	8	((000010(1,0),011111(15,1)), ((001000(4,0),010101(10,1)), ((100010(1,2),111111(15,3)), ((101000(4,2),110101(10,3))
011111	4	((001011(5,1),010100(10,0)), ((101011(5,3),110100(10,2))
100010	8	((000001(0,1),100011(1,3)), ((000110(3,0),100100(2,2)), ((001001(4,1),101011(5,3)), ((010001(8,1),110011(9,3))
100011	4	((000010(1,0),100001(0,3)), ((011000(12,0),111011(13,3))
100100	4	((010000(8,0),110100(10,2)), ((011000(12,0),111100(14,2))
100110	2	((001111(7,1),101001(4,3))
100111	6	((000001(0,1),100110(3,2)), ((010010(9,0),110101(10,3)), ((010101(10,1),110010(9,2))
101000	10	((001110(7,0),100110(3,2)), ((010011(9,1),111011(13,3)), ((010101(10,1),111101(14,3)), ((010111(11,1),111111(15,3)), ((011101(14,1),110101(10,3))
101001	4	((000100(2,0),101101(6,3)), ((011001(12,1),110000(8,2))
101010	8	((000101(2,1),101111(7,3)), ((000111(3,1),101101(6,3)), ((001101(6,1),100111(3,3)), ((011010(13,0),110000(8,2))
101011	2	((001111(7,1),100100(2,2))
.....	..	...

Table A.22: S-Box 8: $S_8(z) = S_8(z \oplus \delta)$

δ	#z	$(z_1,z_1 \oplus \delta)$, $(z_2,z_2 \oplus \delta)$, ...
.....	..	...
101100	2	((000010(1,0),101110(7,2))
101101	8	((001101(6,1),100000(0,2)), ((001110(7,0),100011(1,3)), ((010100(10,0),111001(12,3)), ((011010(13,0),110111(11,3))
101110	6	((001100(6,0),100010(1,2)), ((010100(10,0),111010(13,2)), ((011001(12,1),110111(11,3))
101111	8	((000000(0,0),101111(7,3)), ((000110(3,0),101001(4,3)), ((010011(9,1),111100(14,2)), ((011100(14,0),110011(9,3))
110001	4	((001011(5,1),111010(13,2)), ((011111(15,1),101110(7,2))
110010	6	((000011(1,1),110001(8,3)), ((001010(5,0),111000(12,2)), ((001011(5,1),111001(12,3))
110011	6	((000101(2,1),110110(11,2)), ((001100(6,0),111111(15,3)), ((010110(11,0),100101(2,3))
110101	6	((001000(4,0),111101(14,3)), ((010111(11,1),100010(1,2)), ((011101(14,1),101000(4,2))
110110	4	((000000(0,0),110110(11,2)), ((011100(14,0),101010(5,2))
110111	2	((011011(13,1),101100(6,2))
111001	4	((000111(3,1),111110(15,2)), ((011110(15,0),100111(3,3))
111010	8	((000100(2,0),111110(15,2)), ((001000(4,0),110010(9,2)), ((010010(9,0),101000(4,2)), ((010110(11,0),101100(6,2))
111011	8	((000011(1,1),111000(12,2)), ((001010(5,0),110001(8,3)), ((010000(8,0),101011(5,3)), ((010001(8,1),101010(5,2))
111101	2	((001001(4,1),110100(10,2))
111110	6	((011011(13,1),100101(2,3)), ((011110(15,0),100000(0,2)), ((011111(15,1),100001(0,3))

Table A.23: S-Box 8: $S_8(z) = S_8(z \oplus \delta)$

A.3 S-Box △-tables

The following eight Δ-tables list all existing 18 bit differentials Δ of the eight S-Box triples $1,2,3$ to $8,1,2$, which comply with the bit mask $\Delta = 00x_1x_2vwvwx_3x_4yzyzx_5x_600$, with $x_i, v, w, y, z \in \{0,1\}$ (see Chapter 3.4). Each $\Delta = \delta_i|\delta_j|\delta_k$ represents a concatenation of three 6 bit differentials $\delta_i, \delta_j, \delta_k$ corresponding to S-Boxes i, j, k (see Appendix A.2). Moreover, the number of 18 bit S-Box triple inputs z, which will cause a collision $f(x_{ex}) = f(x_{ex} \oplus \Delta)$ with $z = x_{ex} \oplus k$ for a particular differential Δ, is given. In general, the greater this number the less random encryptions are required to detect a first collision. Also, those differentials Δ are marked, which are collision resistent, i.e. which can be used to force collisions for any possible 18 bit key k (over all 256 key sets K_i).

i	$\Delta_i = \delta_{i,1}\|\delta_{i,2}\|\delta_{i,3}$	$\#z_1 \cdot \#z_2 \cdot \#z_3 = \|Z_{\Delta_i}\|$	collision resistent
1	000011110010100100	$14 \cdot 8 \cdot 4 = 448$	
2	000011110010101000	$14 \cdot 8 \cdot 6 = 672$	
3	000011110010101100	$14 \cdot 8 \cdot 10 = 1120$	√
4	000011110110100100	$14 \cdot 8 \cdot 4 = 448$	
5	000011110110101000	$14 \cdot 8 \cdot 6 = 672$	
6	000011110110101100	$14 \cdot 8 \cdot 10 = 1120$	√
7	000011111010100100	$14 \cdot 4 \cdot 4 = 224$	
8	000011111010101000	$14 \cdot 4 \cdot 6 = 336$	
9	000011111010101100	$14 \cdot 4 \cdot 10 = 560$	√
10	000011111110100100	$14 \cdot 4 \cdot 4 = 224$	
11	000011111110101000	$14 \cdot 4 \cdot 6 = 336$	
12	000011111110101100	$14 \cdot 4 \cdot 10 = 560$	√
13	000111110010100100	$2 \cdot 8 \cdot 4 = 64$	
14	000111110010101000	$2 \cdot 8 \cdot 6 = 96$	
15	000111110010101100	$2 \cdot 8 \cdot 10 = 160$	√
16	000111110110100100	$2 \cdot 8 \cdot 4 = 64$	
17	000111110110101000	$2 \cdot 8 \cdot 6 = 96$	
18	000111110110101100	$2 \cdot 8 \cdot 10 = 160$	√
19	000111111010100100	$2 \cdot 4 \cdot 4 = 32$	
20	000111111010101000	$2 \cdot 4 \cdot 6 = 48$	
21	000111111010101100	$2 \cdot 4 \cdot 10 = 80$	
22	000111111110100100	$2 \cdot 4 \cdot 4 = 32$	
23	000111111110101000	$2 \cdot 4 \cdot 6 = 48$	
24	000111111110101100	$2 \cdot 4 \cdot 10 = 80$	
..	.....	.....	.

Table A.24: S-Box triple 1,2,3

i	$\Delta_i = \delta_{i,1}\|\delta_{i,2}\|\delta_{i,3}$	$\#z_1 \cdot \#z_2 \cdot \#z_3 = \|Z_{\Delta_i}\|$	collision resistent
..	.....	.....	.
25	001011110010100100	$2 \cdot 8 \cdot 4 = 64$	
26	001011110010101000	$2 \cdot 8 \cdot 6 = 96$	
27	001011110010101100	$2 \cdot 8 \cdot 10 = 160$	$\surd$
28	001011110110100100	$2 \cdot 8 \cdot 4 = 64$	
29	001011110110101000	$2 \cdot 8 \cdot 6 = 96$	
30	001011110110101100	$2 \cdot 8 \cdot 10 = 160$	$\surd$
31	001011111010100100	$2 \cdot 4 \cdot 4 = 32$	
32	001011111010101000	$2 \cdot 4 \cdot 6 = 48$	
33	001011111010101100	$2 \cdot 4 \cdot 10 = 80$	
34	001011111110100100	$2 \cdot 4 \cdot 4 = 32$	
35	001011111110101000	$2 \cdot 4 \cdot 6 = 48$	
36	001011111110101100	$2 \cdot 4 \cdot 10 = 80$	
37	001111110010100100	$2 \cdot 8 \cdot 4 = 64$	
38	001111110010101000	$2 \cdot 8 \cdot 6 = 96$	
39	001111110010101100	$2 \cdot 8 \cdot 10 = 160$	$\surd$
40	001111110110100100	$2 \cdot 8 \cdot 4 = 64$	
41	001111110110101000	$2 \cdot 8 \cdot 6 = 96$	
42	001111110110101100	$2 \cdot 8 \cdot 10 = 160$	$\surd$
43	001111111010100100	$2 \cdot 4 \cdot 4 = 32$	
44	001111111010101000	$2 \cdot 4 \cdot 6 = 48$	
45	001111111010101100	$2 \cdot 4 \cdot 10 = 80$	
46	001111111110100100	$2 \cdot 4 \cdot 4 = 32$	
47	001111111110101000	$2 \cdot 4 \cdot 6 = 48$	
48	001111111110101100	$2 \cdot 4 \cdot 10 = 80$	

Table A.25: S-Box triple 1,2,3

i	$\Delta_i = \delta_{i,1}\|\delta_{i,2}\|\delta_{i,3}$	$\#z_1 \cdot \#z_2 \cdot \#z_3 = \|Z_{\Delta_i}\|$	collision resistent
1	000011110010101000	$4 \cdot 2$ cdot $16 = 128$	
2	000011110010101100	$4 \cdot 2 \cdot 8 = 64$	
3	000011110110101000	$4 \cdot 4 \cdot 16 = 256$	$\surd$
4	000011110110101100	$4 \cdot 4 \cdot 8 = 128$	
5	000011111010101000	$4 \cdot 6 \cdot 16 = 384$	
6	000011111010101100	$4 \cdot 6 \cdot 8 = 192$	
7	000011111110101000	$4 \cdot 8 \cdot 16 = 512$	$\surd$
8	000011111110101100	$4 \cdot 8 \cdot 8 = 256$	
9	000111110010101000	$4 \cdot 2 \cdot 16 = 128$	$\surd$
10	000111110010101100	$4 \cdot 2 \cdot 8 = 64$	
11	000111110110101000	$4 \cdot 4 \cdot 16 = 256$	$\surd$
12	000111110110101100	$4 \cdot 4 \cdot 8 = 128$	
13	000111111010101000	$4 \cdot 6 \cdot 16 = 384$	$\surd$
14	000111111010101100	$4 \cdot 6 \cdot 8 = 192$	$\surd$
15	000111111110101000	$4 \cdot 8 \cdot 16 = 512$	$\surd$
16	000111111110101100	$4 \cdot 8 \cdot 8 = 256$	$\surd$
17	001011110010101000	$6 \cdot 2 \cdot 16 = 192$	$\surd$
18	001011110010101100	$6 \cdot 2 \cdot 8 = 96$	
19	001011110110101000	$6 \cdot 4 \cdot 16 = 384$	$\surd$
20	001011110110101100	$6 \cdot 4 \cdot 8 = 192$	
21	001011111010101000	$6 \cdot 6 \cdot 16 = 576$	$\surd$
22	001011111010101100	$6 \cdot 6 \cdot 8 = 288$	$\surd$
23	001011111110101000	$6 \cdot 8 \cdot 16 = 768$	$\surd$
24	001011111110101100	$6 \cdot 8 \cdot 8 = 384$	$\surd$

Table A.26: S-Box triple 2,3,4

i	$\Delta_i = \delta_{i,1}\|\delta_{i,2}\|\delta_{i,3}$	$\#z_1 \cdot \#z_2 \cdot \#z_3 = \|Z_{\Delta_i}\|$	collision resistent
1	000011110010100100	$8 \cdot 16 \cdot 6 = 768$	$\surd$
2	000011110010101000	$8 \cdot 16 \cdot 2 = 256$	$\surd$
3	000011110010101100	$8 \cdot 16 \cdot 8 = 1024$	$\surd$
4	000111110010100100	$2 \cdot 16 \cdot 6 = 192$	$\surd$
5	000111110010101000	$2 \cdot 16 \cdot 2 = 64$	$\surd$
6	000111110010101100	$2 \cdot 16 \cdot 8 = 256$	$\surd$
7	001011110010100100	$2 \cdot 16 \cdot 6 = 192$	$\surd$
8	001011110010101000	$2 \cdot 16 \cdot 2 = 64$	$\surd$
9	001011110010101100	$2 \cdot 16 \cdot 8 = 256$	$\surd$
10	001111110010100100	$4 \cdot 16 \cdot 6 = 384$	$\surd$
11	001111110010101000	$4 \cdot 16 \cdot 2 = 128$	$\surd$
12	001111110010101100	$4 \cdot 16 \cdot 8 = 512$	$\surd$

Table A.27: S-Box triple 3,4,5

i	$\Delta_i = \delta_{i,1}\|\delta_{i,2}\|\delta_{i,3}$	$\#z_1 \cdot \#z_2 \cdot \#z_3 = \|Z_{\Delta_i}\|$	collision resistent
1	000011110010100100	$8 \cdot 4 \cdot 10 = 320$	$\surd$
2	000011110010101000	$8 \cdot 4 \cdot 6 = 192$	$\surd$
3	000011110010101100	$8 \cdot 4 \cdot 4 = 128$	
4	000011110110100100	$8 \cdot 4 \cdot 10 = 320$	
5	000011110110101000	$8 \cdot 4 \cdot 6 = 192$	
6	000011110110101100	$8 \cdot 4 \cdot 4 = 128$	
7	000011111110100100	$8 \cdot 4 \cdot 10 = 320$	
8	000011111110101000	$8 \cdot 4 \cdot 6 = 192$	
9	000011111110101100	$8 \cdot 4 \cdot 4 = 128$	
10	000111110010100100	$4 \cdot 4 \cdot 10 = 160$	$\surd$
11	000111110010101000	$4 \cdot 4 \cdot 6 = 96$	$\surd$
12	000111110010101100	$4 \cdot 4 \cdot 4 = 64$	
13	000111110110100100	$4 \cdot 4 \cdot 10 = 160$	
14	000111110110101000	$4 \cdot 4 \cdot 6 = 96$	
15	000111110110101100	$4 \cdot 4 \cdot 4 = 64$	
16	000111111110100100	$4 \cdot 4 \cdot 10 = 160$	
17	000111111110101000	$4 \cdot 4 \cdot 6 = 96$	
18	000111111110101100	$4 \cdot 4 \cdot 4 = 64$	

Table A.28: S-Box triple 4,5,6

i	$\Delta_i = \delta_{i,1}\|\delta_{i,2}\|\delta_{i,3}$	$\#z_1 \cdot \#z_2 \cdot \#z_3 = \|Z_{\Delta_i}\|$	collision resistent
1	000011110010100100	$8 \cdot 4 \cdot 4 = 128$	
2	000011110010101000	$8 \cdot 4 \cdot 14 = 448$	√
3	000011110010101100	$8 \cdot 4 \cdot 2 = 64$	
4	000011111010100100	$8 \cdot 8 \cdot 4 = 256$	
5	000011111010101000	$8 \cdot 8 \cdot 14 = 896$	√
6	000011111010101100	$8 \cdot 8 \cdot 2 = 128$	
7	000011111110100100	$8 \cdot 4 \cdot 4 = 128$	
8	000011111110101000	$8 \cdot 4 \cdot 14 = 448$	√
9	000011111110101100	$8 \cdot 4 \cdot 2 = 64$	
10	000111110010100100	$2 \cdot 4 \cdot 4 = 32$	
11	000111110010101000	$2 \cdot 4 \cdot 14 = 112$	
12	000111110010101100	$2 \cdot 4 \cdot 2 = 16$	
13	000111111010100100	$2 \cdot 8 \cdot 4 = 64$	
14	000111111010101000	$2 \cdot 8 \cdot 14 = 224$	√
15	000111111010101100	$2 \cdot 8 \cdot 2 = 32$	
16	000111111110100100	$2 \cdot 4 \cdot 4 = 32$	
17	000111111110101000	$2 \cdot 4 \cdot 14 = 112$	√
18	000111111110101100	$2 \cdot 4 \cdot 2 = 16$	
19	001011110010100100	$4 \cdot 4 \cdot 4 = 64$	
20	001011110010101000	$4 \cdot 4 \cdot 14 = 224$	√
21	001011110010101100	$4 \cdot 4 \cdot 2 = 32$	
22	001011111010100100	$4 \cdot 8 \cdot 4 = 128$	
23	001011111010101000	$4 \cdot 8 \cdot 14 = 448$	√
24	001011111010101100	$4 \cdot 8 \cdot 2 = 64$	
25	001011111110100100	$4 \cdot 4 \cdot 4 = 64$	
26	001011111110101000	$4 \cdot 4 \cdot 14 = 224$	√
27	001011111110101100	$4 \cdot 4 \cdot 2 = 32$	
28	001111110010100100	$4 \cdot 4 \cdot 4 = 64$	
29	001111110010101000	$4 \cdot 4 \cdot 14 = 224$	√
30	001111110010101100	$4 \cdot 4 \cdot 2 = 32$	
31	001111111010100100	$4 \cdot 8 \cdot 4 = 128$	
32	001111111010101000	$4 \cdot 8 \cdot 14 = 448$	√
33	001111111010101100	$4 \cdot 8 \cdot 2 = 64$	
34	001111111110100100	$4 \cdot 4 \cdot 4 = 64$	
35	001111111110101000	$4 \cdot 4 \cdot 14 = 224$	√
36	001111111110101100	$4 \cdot 4 \cdot 2 = 32$	

Table A.29: S-Box triple 5,6,7

i	$\Delta_i = \delta_{i,1}\|\delta_{i,2}\|\delta_{i,3}$	$\#z_1 \cdot \#z_2 \cdot \#z_3 = \|Z_{\Delta_i}\|$	collision resistent
1	000111110010100100	$6 \cdot 2 \cdot 4 = 48$	
2	000111110010101000	$6 \cdot 2 \cdot 10 = 120$	$\surd$
3	000111110010101100	$6 \cdot 2 \cdot 2 = 24$	
4	000111110110100100	$6 \cdot 2 \cdot 4 = 48$	
5	000111110110101000	$6 \cdot 2 \cdot 10 = 120$	$\surd$
6	000111110110101100	$6 \cdot 2 \cdot 2 = 24$	
7	000111111010100100	$6 \cdot 16 \cdot 4 = 384$	$\surd$
8	000111111010101000	$6 \cdot 16 \cdot 10 = 960$	$\surd$
9	000111111010101100	$6 \cdot 16 \cdot 2 = 192$	$\surd$
10	000111111110100100	$6 \cdot 4 \cdot 4 = 96$	$\surd$
11	000111111110101000	$6 \cdot 4 \cdot 10 = 240$	$\surd$
12	000111111110101100	$6 \cdot 4 \cdot 2 = 48$	$\surd$
13	001011110010100100	$4 \cdot 2 \cdot 4 = 32$	
14	001011110010101000	$4 \cdot 2 \cdot 10 = 80$	
15	001011110010101100	$4 \cdot 2 \cdot 2 = 16$	
16	001011110110100100	$4 \cdot 2 \cdot 4 = 32$	
17	001011110110101000	$4 \cdot 2 \cdot 10 = 80$	
18	001011110110101100	$4 \cdot 2 \cdot 2 = 16$	
19	001011111010100100	$4 \cdot 16 \cdot 4 = 256$	$\surd$
20	001011111010101000	$4 \cdot 16 \cdot 10 = 640$	$\surd$
21	001011111010101100	$4 \cdot 16 \cdot 2 = 128$	$\surd$
22	001011111110100100	$4 \cdot 4 \cdot 4 = 64$	
23	001011111110101000	$4 \cdot 4 \cdot 10 = 160$	
24	001011111110101100	$4 \cdot 4 \cdot 2 = 32$	
25	001111110010100100	$2 \cdot 2 \cdot 4 = 16$	
26	001111110010101000	$2 \cdot 2 \cdot 10 = 40$	
27	001111110010101100	$2 \cdot 2 \cdot 2 = 8$	
28	001111110110100100	$2 \cdot 2 \cdot 4 = 16$	
29	001111110110101000	$2 \cdot 2 \cdot 10 = 40$	
30	001111110110101100	$2 \cdot 2 \cdot 2 = 8$	
31	001111111010100100	$2 \cdot 16 \cdot 4 = 128$	$\surd$
32	001111111010101000	$2 \cdot 16 \cdot 10 = 320$	$\surd$
33	001111111010101100	$2 \cdot 16 \cdot 2 = 64$	$\surd$
34	001111111110100100	$2 \cdot 4 \cdot 4 = 32$	
35	001111111110101000	$2 \cdot 4 \cdot 10 = 80$	
36	001111111110101100	$2 \cdot 4 \cdot 2 = 16$	

Table A.30: S-Box triple 6,7,8

i	$\Delta_i = \delta_{i,1}\|\delta_{i,2}\|\delta_{i,3}$	$\#z_1 \cdot \#z_2 \cdot \#z_3 = \|Z_{\Delta_i}\|$	collision resistent
1	000011110010100100	$8 \cdot 6 \cdot 12 = 576$	$\surd$
2	000011110010101000	$8 \cdot 6 \cdot 12 = 576$	$\surd$
3	000011110010101100	$8 \cdot 6 \cdot 4 = 192$	$\surd$
4	000011110110100100	$8 \cdot 4 \cdot 12 = 384$	$\surd$
5	000011110110101000	$8 \cdot 4 \cdot 12 = 384$	$\surd$
6	000011110110101100	$8 \cdot 4 \cdot 4 = 128$	
7	000011111010100100	$8 \cdot 8 \cdot 12 = 768$	$\surd$
8	000011111010101000	$8 \cdot 8 \cdot 12 = 768$	$\surd$
9	000011111010101100	$8 \cdot 8 \cdot 4 = 256$	
10	000011111110100100	$8 \cdot 6 \cdot 12 = 576$	$\surd$
11	000011111110101000	$8 \cdot 6 \cdot 12 = 576$	$\surd$
12	000011111110101100	$8 \cdot 6 \cdot 4 = 192$	$\surd$
13	000111110010100100	$4 \cdot 6 \cdot 12 = 288$	$\surd$
14	000111110010101000	$4 \cdot 6 \cdot 12 = 288$	$\surd$
15	000111110010101100	$4 \cdot 6 \cdot 4 = 96$	$\surd$
16	000111110110100100	$4 \cdot 4 \cdot 12 = 192$	$\surd$
17	000111110110101000	$4 \cdot 4 \cdot 12 = 192$	$\surd$
18	000111110110101100	$4 \cdot 4 \cdot 4 = 64$	
19	000111111010100100	$4 \cdot 8 \cdot 12 = 384$	$\surd$
20	000111111010101000	$4 \cdot 8 \cdot 12 = 384$	$\surd$
21	000111111010101100	$4 \cdot 8 \cdot 4 = 128$	
22	000111111110100100	$4 \cdot 6 \cdot 12 = 288$	$\surd$
23	000111111110101000	$4 \cdot 6 \cdot 12 = 288$	$\surd$
24	000111111110101100	$4 \cdot 6 \cdot 4 = 96$	$\surd$
25	001111110010100100	$2 \cdot 6 \cdot 12 = 144$	
26	001111110010101000	$2 \cdot 6 \cdot 12 = 144$	
27	001111110010101100	$2 \cdot 6 \cdot 4 = 48$	
28	001111110110100100	$2 \cdot 4 \cdot 12 = 96$	$\surd$
29	001111110110101000	$2 \cdot 4 \cdot 12 = 96$	$\surd$
30	001111110110101100	$2 \cdot 4 \cdot 4 = 32$	
31	001111111010100100	$2 \cdot 8 \cdot 12 = 192$	$\surd$
32	001111111010101000	$2 \cdot 8 \cdot 12 = 192$	$\surd$
33	001111111010101100	$2 \cdot 8 \cdot 4 = 64$	
34	001111111110100100	$2 \cdot 6 \cdot 12 = 144$	
35	001111111110101000	$2 \cdot 6 \cdot 12 = 144$	
36	001111111110101100	$2 \cdot 6 \cdot 4 = 48$	

Table A.31: S-Box triple 7,8,1

i	$\Delta_i = \delta_{i,1}\|\delta_{i,2}\|\delta_{i,3}$	$\#z_1 \cdot \#z_2 \cdot \#z_3 = \|Z_{\Delta_i}\|$	collision resistent
1	000011110010100100	$6 \cdot 4 \cdot 10 = 240$	√
2	000011110010101000	$6 \cdot 4 \cdot 8 = 192$	√
3	000011110010101100	$6 \cdot 4 \cdot 10 = 240$	√
4	000011110110100100	$6 \cdot 2 \cdot 10 = 120$	√
5	000011110110101000	$6 \cdot 2 \cdot 8 = 96$	√
6	000011110110101100	$6 \cdot 2 \cdot 10 = 120$	√
7	000011111010100100	$6 \cdot 6 \cdot 10 = 360$	√
8	000011111010101000	$6 \cdot 6 \cdot 8 = 288$	√
9	000011111010101100	$6 \cdot 6 \cdot 10 = 360$	√
10	000011111110100100	$6 \cdot 4 \cdot 10 = 240$	√
11	000011111110101000	$6 \cdot 4 \cdot 8 = 192$	√
12	000011111110101100	$6 \cdot 4 \cdot 10 = 240$	√
13	000111110010100100	$6 \cdot 4 \cdot 10 = 240$	√
14	000111110010101000	$6 \cdot 4 \cdot 8 = 192$	√
15	000111110010101100	$6 \cdot 4 \cdot 10 = 240$	√
16	000111110110100100	$6 \cdot 2 \cdot 10 = 120$	
17	000111110110101000	$6 \cdot 2 \cdot 8 = 96$	
18	000111110110101100	$6 \cdot 2 \cdot 10 = 120$	
19	000111111010100100	$6 \cdot 6 \cdot 10 = 360$	√
20	000111111010101000	$6 \cdot 6 \cdot 8 = 288$	√
21	000111111010101100	$6 \cdot 6 \cdot 10 = 360$	√
22	000111111110100100	$6 \cdot 4 \cdot 10 = 240$	√
23	000111111110101000	$6 \cdot 4 \cdot 8 = 192$	√
24	000111111110101100	$6 \cdot 4 \cdot 10 = 240$	√
25	001011110010100100	$2 \cdot 4 \cdot 10 = 80$	√
26	001011110010101000	$2 \cdot 4 \cdot 8 = 64$	√
27	001011110010101100	$2 \cdot 4 \cdot 10 = 80$	√
28	001011110110100100	$2 \cdot 2 \cdot 10 = 40$	
29	001011110110101000	$2 \cdot 2 \cdot 8 = 32$	
30	001011110110101100	$2 \cdot 2 \cdot 10 = 40$	
31	001011111010100100	$2 \cdot 6 \cdot 10 = 120$	√
32	001011111010101000	$2 \cdot 6 \cdot 8 = 96$	√
33	001011111010101100	$2 \cdot 6 \cdot 10 = 120$	√
34	001011111110100100	$2 \cdot 4 \cdot 10 = 80$	√
35	001011111110101000	$2 \cdot 4 \cdot 8 = 64$	√
36	001011111110101100	$2 \cdot 4 \cdot 10 = 80$	√
37	001111110010100100	$8 \cdot 4 \cdot 10 = 320$	√
38	001111110010101000	$8 \cdot 4 \cdot 8 = 256$	√
39	001111110010101100	$8 \cdot 4 \cdot 10 = 320$	√
40	001111110110100100	$8 \cdot 2 \cdot 10 = 160$	
41	001111110110101000	$8 \cdot 2 \cdot 8 = 128$	
42	001111110110101100	$8 \cdot 2 \cdot 10 = 160$	
43	001111111010100100	$8 \cdot 6 \cdot 10 = 480$	√
44	001111111010101000	$8 \cdot 6 \cdot 8 = 384$	√
45	001111111010101100	$8 \cdot 6 \cdot 10 = 480$	√
46	001111111110100100	$8 \cdot 4 \cdot 10 = 320$	√
47	001111111110101000	$8 \cdot 4 \cdot 8 = 256$	√
48	001111111110101100	$8 \cdot 4 \cdot 10 = 320$	√

Table A.32: S-Box triple 8,1,2

Bibliography

[AK96] R. Anderson and M. Kuhn. Tamper Resistance - a Cautionary Note. In *Second Usenix Workshop on Electronic Commerce*, pages 1–11, November 1996.

[AO] M. Aigner and E. Oswald. Power Analysis Tutorial. http://www.iaik.tugraz.at/aboutus/people/oswald/papers/dpa_tutorial.pdf. Seminar paper.

[BGW98a] M. Briceno, I. Goldberg, and D. Wagner. An Implementation of the GSM A3A8 algorithm, 1998. http://www.scard.org/gsm/a3a8.txt.

[BGW98b] M. Briceno, I. Goldberg, and D. Wagner. GSM cloning, 1998. http://www.isaac.cs.berkely.edu/isaac/gsm–faq.html.

[BS91] E. Biham and A. Shamir. Differential Cryptanalysis of DES-like Cryptosystems. In A. J. Menezes and S. A. Vanstone, editors, *Advances in Cryptology — CRYPTO '90*, volume LNCS 537, pages 2–21, Berlin, Germany, 1991. Springer-Verlag.

[CCD00] C. Clavier, J.S. Coron, and N. Dabbous. Differential Power Analysis in the Presence of Hardware Countermeasures. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2000*, volume LNCS 1965, pages 252–263. Springer-Verlag, 2000.

[Cop94] D. Coppersmith. The Data Encryption Standard (DES) and its Strength Against Attacks. Technical report rc 186131994, IBM Thomas J. Watson Research Center, December 1994.

[Dob01] H. Dobbertin. Mathematische Grundlagen der Kryptographie, 2001. Lecture notes spring semester.

[KJJ98] P. Kocher, J. Jaffe, and B. Jun. Introduction to Differential Power Analysis and Related Attacks. http://www.cryptography.com/dpa/technical, 1998. Manuscript, Cryptography Research, Inc.

[KJJ99] P. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In *Advances in Cryptology — CRYPTO '99*, volume LNCS 1666, pages 388–397. Springer-Verlag, 1999.

[Koc96] P. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *Advances in Cryptology — CRYPTO '96*, volume LNCS 1666, pages 104–113. Springer-Verlag, 1996.

[LJ01] P. Laackmann and M. Janke. Power- und Timing-Analysen: Angriffe auf geheime Kartendaten. *Zeitschrift für Sicherheit in der Wirtschaft*, December 2001.

[Mat94] M. Matsui. Linear Cryptanalysis of DES Cipher. In T. Hellenseth, editor, *Advances in Cryptology — EUROCRYPT '93*, volume LNCS 0765, pages 286 – 397, Berlin, Germany, 1994. Springer-Verlag.

[MDS99] T. S. Messerges, E. A. Dabbish, and R. H. Sloan. Investigations of Power Analysis Attacks on Smartcards. In *USENIX Workshop on Smartcard Technology*, pages 151–162, 1999.

[MS00] R. Mayer-Sommer. Smartly Analyzing the Simplicity and the Power of Simple Power Analysis on Smart Cards. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2000*, volume LNCS 1965, pages 78 – 92. Springer-Verlag, 2000.

[Mui01] J.A. Muir. Techniques of Side Channel Cryptanalysis. Master thesis, 2001. University of Waterloo, Canada.

[MvOV97] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, Florida, USA, 1997.

[Oli00] F. Olivier. Blind Deconvolution and DPA Resynchronization. Proceedings in CHES 2000, 2000. presentation.

[Paa02] C. Paar. Fundamentals of Cryptography, 2002. Lecture notes spring semester.

[Pac93] Hewlett Packard. *HP 1660 Series Logic Analyzers: Programmer's Guide*, 1993.

[RSA] RSA security. DES Challenge. http://www.rsasecurity.com/.

[Sch96] B. Schneier. *Applied Cryptography*. John Wiley & Sons, 2nd edition edition, 1996.

[Sel02] M. Selhorst. Die Geldkarte - Eine sichere elektronische Geldbörse?! Seminar paper, 2002. Universität Bochum, Germany.

[Ser] Vault Information Services. 8051 online tuorials, source codes, etc. www.8052.com. .

[Sti95] D. R. Stinson. *Cryptography: Theory and Practice*. CRC Press, Boca Raton, Florida, USA, 1995.

[Tec01a] Agilent Technologies. *82357A USB/GPIB Interface Standard Instrument Control Library*, 2001.

[Tec01b] Infineon Technologies. Security & Chip Card ICs SLE 66CX320P. Preliminary Short Product Information, August 2001. Munich, Germany.

[WE93] N. Weste and K. Eshraghian. *Principles of CMOS VLSI Design*. Addison-Wesley Publishing Company, 1993.

[Zen99] E. Zenner. Kryptographische Protokolle im GSM-Standard: Beschreibung und Kryptanalyse. Master thesis, 1999. Universität Mannheim, Germany.